

Progetto di una Basi di Dati  
Circolo Scolastico  
Università degli Studi di Perugia  
Corso di Laurea in Ingegneria Informatica ed  
Elettronica  
A.A. 2021/2022  
Docente: Giuseppe Liotta

Gian Marco Ferri 314666



## Indice

<b>1</b>	<b>Progettazione Concettuale</b>	<b>3</b>
1.1	Raccolta dei requisiti . . . . .	3
1.2	Analisi delle specifiche . . . . .	5
1.3	Glossario dei termini principali . . . . .	7
1.4	Elenco delle operazioni . . . . .	8
1.5	Schema Entity-Relationship . . . . .	10
1.6	Vincoli non esprimibili . . . . .	16
1.7	Dizionario dei dati (Entità) . . . . .	17
1.8	Dizionario dei dati (Relazioni) . . . . .	20
<b>2</b>	<b>Progettazione Logica</b>	<b>21</b>
2.1	Tavola dei volumi . . . . .	21
2.2	Tavola delle operazioni . . . . .	23
2.3	Analisi delle ridondanze . . . . .	24
2.4	Eliminazione delle generalizzazioni . . . . .	32
2.5	Partizionamento di Entità e Relazioni . . . . .	38
2.6	Accorpamento di Entità e Relazioni . . . . .	41
2.7	Scelta degli identificatori primari . . . . .	42
2.8	Modello relazionale . . . . .	44
2.8.1	Traduzione di Entità . . . . .	44
2.8.2	Traduzione di Relazioni . . . . .	46
<b>3</b>	<b>Implementazione delle operazioni</b>	<b>48</b>
3.1	Creazione delle tabelle . . . . .	48
3.2	Creazione dei vincoli non esprimibili . . . . .	58
3.3	Creazione delle query . . . . .	59
<b>4</b>	<b>Esecuzione delle operazioni principali</b>	<b>80</b>

# 1 Progettazione Concettuale

## 1.1 Raccolta dei requisiti

*Attraverso la raccolta dei requisiti vengono individuate le caratteristiche che la base dati dovrà avere per rispondere alle esigenze del committente.*

Si vuole realizzare una base di dati informativa finalizzata a modellare l'organizzazione del Primo Circolo Scolastico del Comune di Foligno.

Tale Circolo è formato da cinque scuole primarie e cinque scuole dell'infanzia, situate nelle diverse frazioni del comune di Foligno. Questo viene identificato da un nome e da un codice meccanografico.

Le scuole primarie sono suddivise in cinque classi, dalla prima alla quinta. Nelle scuole materne invece non vi è la suddivisione in classi, ma i bambini sono divisi in base all'età. Ogni plesso è individuato dal nome, l'indirizzo (che comprende città, provincia, via, numero civico, CAP), il numero di telefono, l'indirizzo e-mail e il codice meccanografico.

Nella scuola elementare vi lavorano complessivamente 80 insegnanti. Le maestre delle elementari insegnano una o al massimo due materie in più classi. Ad eccezione dei docenti di religione, che seguono molte più classi e hanno associata un'unica materia. Sono presenti maestre di ruolo e supplenti. Quelle di ruolo, in caso di malattia, possono essere sostituite dalle supplenti. Inoltre, vi sono le docenti di sostegno (solo nelle scuole elementari), che si occupano di almeno un alunno avente la legge 104.

Nella scuola materna vi lavorano almeno cinque maestre le quali non hanno associata nessuna materia. Non si rappresentano, in questo caso, maestre di sostegno per bambini aventi disabilità. Si identifica ogni docente, sia di scuola elementare sia d'infanzia, con i suoi dati anagrafici, l'indirizzo e-mail scolastico, il codice insegnante (o supplente), un numero di telefono e il loro relativo stipendio. Quest'ultimo si aggiorna ogni cinque anni, in base agli anni trascorsi a insegnare.

Il personale ATA è composto dai collaboratori scolastici (addetti alle pulizie) dell'edificio scolastico e i collaboratori per il servizio mensa. Di questi si conoscono la scuola in cui lavorano, dati anagrafici, numero telefonico, indirizzo e-mail e stipendio.

Ogni alunno è iscritto a un'unica scuola, un unico anno scolastico e un'unica classe.

Di ogni alunno si conosce un codice identificativo scolastico, i suoi dati anagrafici, quelli dei genitori e, in caso di assenza di quest'ultimi, quelli del tutore a

cui è stato affidato. Inoltre, vengono chiesti ai genitori i documenti e il numero telefonico da chiamare in caso di emergenza. Nella base dati vengono anche memorizzati i voti ottenuti nel corso di studi dell'anno corrente. La valutazione viene espressa in decimi.

Al termine dei cinque anni di elementari, lo studente viene cancellato dal database e con lui tutti i dati relativi ai parenti e/o tutori (come unico caso eccezionale, si ha quello in cui ci sono fratelli nello stesso circolo).

Per quanto riguarda i bambini che frequentano la materna, anche di loro si conoscono i dati anagrafici dei genitori e di tutte le persone autorizzate a riportare il bambino a casa. Ogni scuola materna offre il servizio mensa e nella base dati si vogliono memorizzare anche le allergie dei bambini e le tipologie di vaccinazioni effettuate.

Il Circolo è diretto da un unico dirigente scolastico affiancato da due collaboratori e dalla segreteria. Quest'ultima composta da otto persone, che si occupano della gestione burocratica e amministrativa del Circolo. All'interno della segreteria, si distinguono i segretari e gli applicati della segreteria. Questi ultimi svolgono funzioni di *back office*.

## 1.2 Analisi delle specifiche

*La fase di analisi delle specifiche consiste in un affinamento e riorganizzazione di quanto acquisito durante la raccolta dei requisiti.*

### **Frasi di carattere generale:**

Si vuole realizzare una base di dati informativa finalizzata a modellare l'organizzazione del Circolo Scolastico del Comune di Foligno.

### **Frasi relative al circolo:**

Il circolo è formato da cinque scuole primarie e cinque scuole dell'infanzia, situate nelle diverse frazioni del comune di Foligno. Questo viene identificato da un nome e da un codice meccanografico.

### **Frasi relative al plesso:**

Ogni plesso è individuato da un nome, un indirizzo, un numero di telefono, un indirizzo e-mail e un codice meccanografico.

### **Frasi relative ai docenti di ogni plesso:**

Si identifica ogni docente, sia di scuola elementare sia d'infanzia, con i suoi dati anagrafici, un indirizzo e-mail scolastico, un codice insegnante (o supplente), un numero di telefono e suo relativo stipendio. Quest'ultimo si aggiorna ogni cinque anni, in base agli anni trascorsi a insegnare.

### **Frasi relative alle scuole primarie:**

Le scuole primarie sono suddivise in cinque classi, dalla prima alla quinta.

### **Frasi relative ai docenti delle scuole primarie:**

Le maestre delle elementari insegnano una o al massimo due materie in 2 classi (se la classe ha più sezioni insegnano in tutte le sezioni), a eccezione delle docenti di religione, che seguono molte più classi e hanno associata un'unica materia. Sono presenti maestre di ruolo e supplenti. Quelle di ruolo, in caso di malattia, possono essere sostituite dalle supplenti. Inoltre, vi sono le docenti di sostegno (solo nelle scuole elementari), che si occupano di almeno un alunno avente la legge 104.

### **Frasi relative agli alunni delle scuole primarie:**

Ogni alunno è iscritto a un'unica scuola, un unico anno scolastico e un'unica classe. Gli alunni seguono obbligatoriamente tutte le materie, ad eccezione di IRC (Insegnamento Religione Cattolica), dove, all'inizio di ogni anno, viene data alle famiglie la possibilità di scelta di far frequentare la materia ai propri figli. Al termine dei cinque anni di elementari lo studente viene cancellato dal database e con lui tutti i dati relativi ai parenti e/o tutori (salvo caso in cui si hanno fratelli nella stessa scuola).

**Frase relative ai voti delle materie della scuola primaria:**

Nella base dati vengono anche memorizzati i voti ottenuti nel corso di studi dell'anno corrente. La valutazione viene espressa in decimi.

**Frase relative alle scuole materne:**

Nelle scuole materne non vi è la suddivisione in classi, ma i bambini sono divisi in base all'età. Ogni scuola materna offre il servizio mensa.

**Frase relative ai docenti della scuola materna:**

Nella scuola materna vi lavorano almeno cinque maestre le quali non hanno associata nessuna materia. Non si rappresentano, in questo caso, maestre di sostegno per bambini aventi disabilità.

**Frase relative ai bambini della scuola materna:**

Per quanto riguarda i bambini che frequentano la materna, anche di loro si conoscono i dati anagrafici dei genitori e di tutte le persone autorizzate a riportare il bambino a casa. Si vogliono memorizzare anche le allergie dei bambini e le tipologie di vaccinazioni effettuate.

**Frase relative agli affidatari dei bambini e alunni:**

Di ogni affidatario, si conoscono i dati anagrafici dei genitori e, in caso di assenza di quest'ultimi, quelli del tutore a cui è affidato il bambino o l'alunno. Inoltre, vengono chiesti ai genitori i documenti e il numero telefonico da chiamare in caso di emergenza.

**Frase relative al personale ATA:**

Il personale ATA è composto dai collaboratori scolastici (addetti alle pulizie) dell'edificio scolastico e i collaboratori per il servizio mensa. Di questi si conosce la scuola in cui lavorano, dati anagrafici, numero telefonico, indirizzo e-mail e stipendio. Fanno parte di questa sezione anche i collaboratori del dirigente scolastico e la segreteria. All'interno della segreteria, si distinguono i segretari e gli applicati della segreteria.

**Frase relative alla direzione del Circolo:**

Il Circolo è diretto da un unico dirigente scolastico affiancato da due collaboratori e dalla segreteria. Quest'ultima composta da otto persone, che si occupano della gestione burocratica e amministrativa del Circolo.

### 1.3 Glossario dei termini principali

*Il glossario dei termini semplifica la comprensione delle specifiche, chiarendo in particolar modo i termini centrali della realtà di interesse e di conseguenza anche della base dati.*

Termine	Descrizione	Sinonimi	Collegamenti
Insegnante	Persona che per professione istruisce in una data disciplina	Maestra, Docente	Alunni, Classi, Materia
Plesso	Istituto dove si svolgono le lezioni	Scuola	Maestre, Circolo
Dati Anagrafici	Nome, cognome, sesso, codice fiscale, data di nascita		Maestre, Alunni, Preside
Residenza	Luogo in cui una persona ha la dimora attuale		
Circolo	Insieme delle sedi scolastiche	Istituto	Scuola
Mensa	Servizio destinato alla consumazione dei pasti		Collaboratori, Bambini
Alunni	Frequentante della scuola elementare		
Bambino	Frequentante della scuola materna		
Personale ATA	Personale amministrativo, tecnico e ausiliario (personale non docente che lavora nella scuola italiana)	Addetti alle pulizie, collaboratori, segreteria	
Dirigente scolastico	Dirigente preposto al vertice di una scuola	Preside	Insegnanti, Alunni, Segreteria

## 1.4 Elenco delle operazioni

*L'elenco delle operazioni principali determina la conclusione della prima parte della progettazione concettuale. L'informazione e la frequenza con la quale vengono svolte tali operazioni, sono di fondamentale importanza ai fini della progettazione logica e dell'ottimizzazione dell'informazione da rappresentare.*

- **OP 1** : Iscrivere un nuovo alunno. (250 volte ogni anno scolastico, 50 per ogni scuola elementare);
- **OP 2** : Aggiungere un nuovo bambino. (100 volte ogni anno scolastico per ogni scuola materna);
- **OP 3** : Aggiungere un docente di scuola primaria. (1 volta ogni anno scolastico);
- **OP 4** : Aggiungere un affidatario. (2 volte all'inizio del ciclo scolastico);
- **OP 5** : Aggiungere un dipendente del personale ATA. (1 volta ogni anno scolastico);
- **OP 6** : Assegnare una nuova supplente di scuola elementare. (1 volta ogni anno scolastico);
- **OP 7** : Aggiungere una maestra di ruolo di scuola materna. (1 volta all'anno);
- **OP 8** : Assegnare una valutazione a un alunno. (2 volte ogni anno scolastico per ogni alunno);
- **OP 9** : Eliminare gli alunni che hanno concluso il ciclo scolastico. (250 volte ogni anno scolastico, 50 per ogni scuola elementare);
- **OP 10** : Eliminare una maestra. (1 volta ogni cinque anni);
- **OP 11** : Sostituire la preside. (1 volta ogni cinque anni);
- **OP 12** : Sostituire una docente di ruolo con una supplente. (2 volte a semestre);
- **OP 13** : Inserire le allergie di un bambino già iscritto alla scuola. (1 volta ogni ciclo scolastico per ogni bambino);
- **OP 14** : Assegnare una lezione in presenza o online a una classe. (1 volta ogni settimana per ogni classe);
- **OP 15** : Assegnare una materia al docente. (2 volte ogni anno scolastico per ogni docente di scuola elementare);
- **OP 16** : Aggiornare lo stipendio del personale ATA. (1 volta ogni 5 anni);



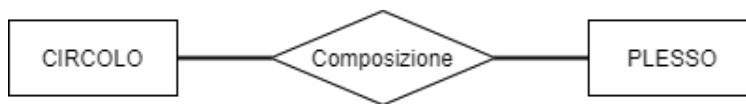
- **OP 17** : Stampare i voti della pagella di un alunno. (2 volte ogni anno scolastico);
- **OP 18** : Stampare i docenti di una classe. (1 volta ogni anno scolastico);
- **OP 19** : Stampare gli alunni di una classe. (1 volta ogni anno scolastico);
- **OP 20** : Stampare i dati lavorativi del personale ATA del circolo. (1 volta ogni anno scolastico);
- **OP 21** : Stampare nome, cognome e codice insegnante di tutti i docenti di un plesso. (1 volta ogni anno per ogni circolo);
- **OP 22** : Stampare i codici insegnante delle docenti del circolo. (1 volta ogni anno per ogni circolo);
- **OP 23** : Stampare nome, cognome, scuola e classe di uno studente. (1.250 volte ogni semestre);
- **OP 24** : Stampare gli studenti che non frequentano IRC. (1 volta ogni anno scolastico);
- **OP 25** : Stampare i dati anagrafici del personale ATA. (1 volta ogni anno scolastico);
- **OP 26** : Stampare tutte le supplenti che hanno sostituito le docenti presenti e/o passate. (1 volta ogni anno scolastico);

## 1.5 Schema Entity-Relationship

*Al fine di proseguire nella progettazione concettuale, si cerca uno schema E-R che descriva al meglio le specifiche precedentemente raccolte ed analizzate.*

Per la realizzazione dello schema E-R , si è adottata una strategia mista. Inizialmente, si è partiti da uno schema scheletro, rappresentante i concetti fondamentali della base di dati. Si sono considerate come entità CIRCOLO e PLESSO. La relazione Composizione, le cui occorrenze rappresentano i plessi (scuole) da cui è composto il circolo, lega appunto CIRCOLO e PLESSO.

Di seguito, viene riportato in figura lo schema scheletro di partenza.

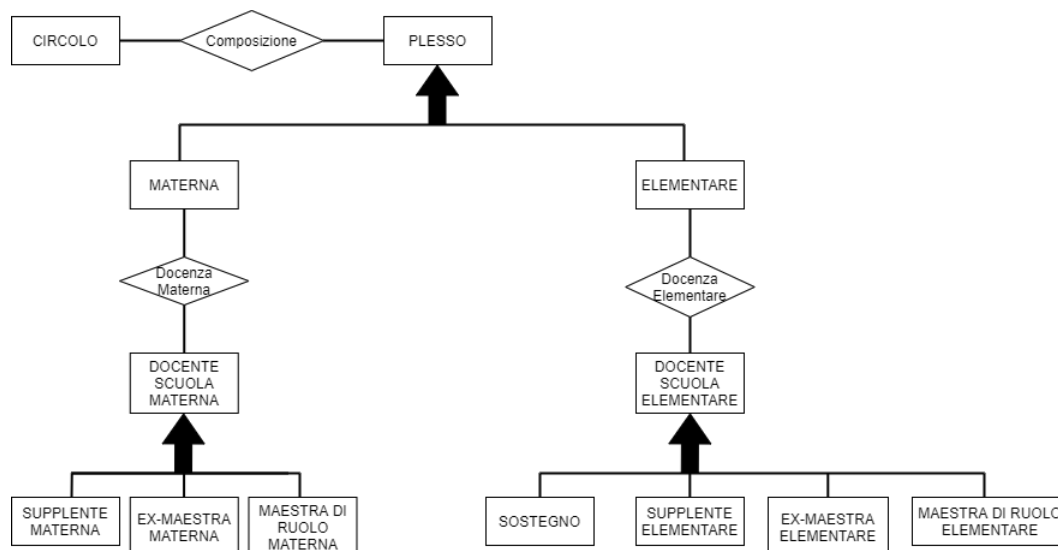


Si procede, nella realizzazione, a macchia d'olio espandendo lo schema di partenza a partire dai concetti definiti. Si introduce una generalizzazione totale ed esclusiva, poiché il PLESSO può essere solamente una scuola MATERNA o una scuola ELEMENTARE dalle specifiche sopra descritte.

A ogni scuola sono associate più maestre, rispettivamente per le scuole elementari e materne. Dunque, viene definita la relazione Docenza Elementare (Materna) tra le entità ELEMENTARE (MATERNA) e DOCENTE SCUOLA ELEMENTARE (MATERNA).

Esse stesse sono generalizzazioni totali ed esclusive delle tipologie di maestre rappresentate nella base di dati: MAESTRA DI RUOLO ELEMENTARE (MATERNA), EX-MAESTRA ELEMENTARE (MATERNA), SUPLENTE ELEMENTARE (MATERNA) e, solo nelle scuole elementari, la maestra di SOSTEGNO.

Si riporta, nella figura sottostante, l'espansione apportata alla schema E-R .

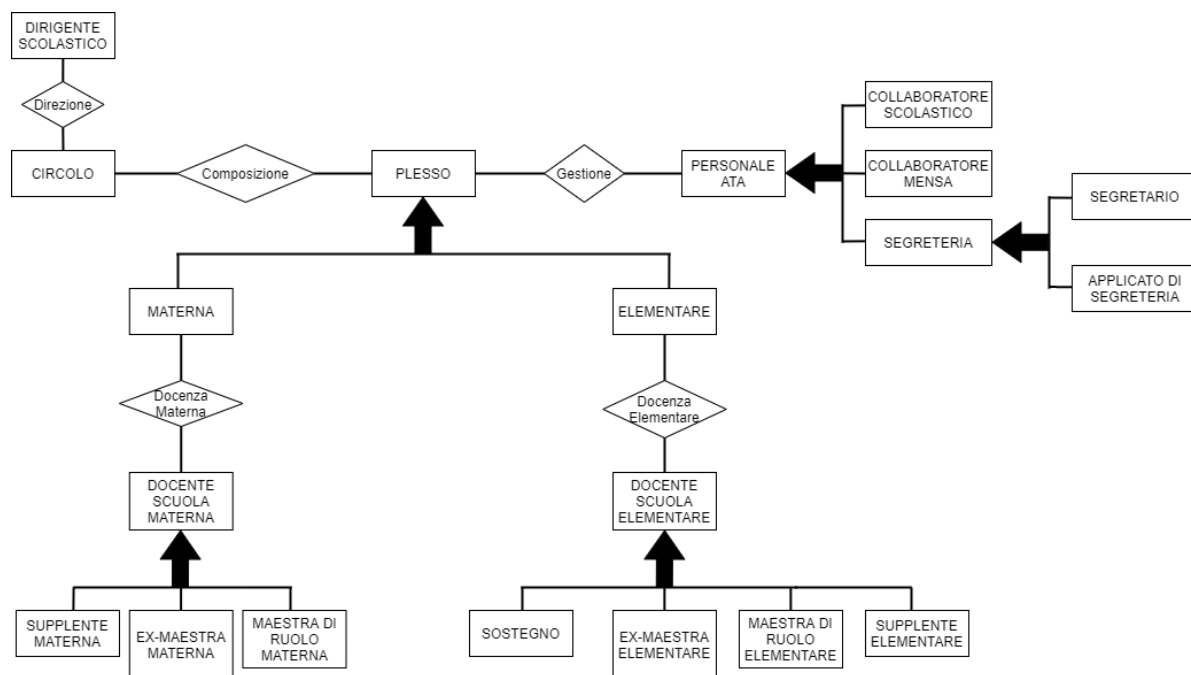


A partire da CIRCOLO si è di nuovo esteso lo schema, considerando ulteriori concetti “vicini”. Da considerare, c’è la zona amministrativa del circolo composta dal PERSONALE ATA e la direzione rappresentata dal DIRIGENTE SCOLASTICO e dai suoi collaboratori (che rientrano comunque nel personale ATA).

Si è definita la relazione Direzione tra CIRCOLO e DIRIGENTE SCOLASTICO, mentre si è associata l’entità PERSONALE ATA al PLESSO tramite la relazione Gestione. PERSONALE ATA è a sua volta generalizzazione totale ed esclusiva delle entità: COLLABORATORE SCOLASTICO, SEGRETERIA e COLLABORATORE MENSA (poiché è previsto il servizio mensa).

Infine, l’entità SEGRETERIA è generalizzazione totale ed esclusiva di SEGRETERARIO e APPLICATO DI SEGRETERIA.

Si riporta di seguito l’estensione apportata allo schema.



Per raffinare ulteriormente lo schema, si considerano le informazioni riguardanti i bambini e gli alunni appartenenti alla scuole.

Viene definita una relazione *Iscrizione Materna* tra *MATERNA* e l'entità *BAMBINO* per rappresentare l'iscrizione a una scuola materna di un numero non definito di bambini. A questa entità è associata un'ulteriore entità, che rappresenta l'*AFFIDATARIO*, cioè la persona a cui viene affidata la responsabilità di recuperare il bambino all'uscita della scuola.

Per realizzare tale collegamento, si è definita la relazione *Affidamento* che è valida non solo per l'entità *BAMBINO*, ma anche per *ALUNNO* (coloro che sono iscritti alla scuola elementare).

Infatti, anch'esso, ha bisogno di un accompagnatore all'uscita. Tra *ELEMENTARE* e *ALUNNO* viene introdotta, analogamente, una relazione *Iscrizione Elementare*.

*AFFIDATARIO* è generalizzazione totale ed esclusiva delle uniche persone ammesse all'uscita, per ritirare il bambino o l'alunno: l'entità *GENITORE* o l'entità *TUTORE*.

Ogni scuola elementare è suddivisa in classi. Risulta necessario aggiungere una relazione che ad ogni occorrenza di *ELEMENTARE* assegni una o più occorrenze di *CLASSE*: si introduce la relazione *Divisione*.

Inoltre, ogni alunno della scuola elementare (materna), è iscritto ad un'unica scuola; ciò è rappresentato dalla relazione *Iscrizione Elementare (Materna)*.

Un'ulteriore parte da raffinare è quella associata alle docenti delle scuole elementari e agli alunni. Dalle specifiche del *database*, emerge che una maestra può avere (o aver avuto) più docenze in diverse classi, anche di materie diverse.

Per questo si è introdotta la relazione Insegnamento Materia, che definisce quanto appena esplicitato. Questa connette l'entità MAESTRA DI RUOLO ELEMENTARE alla materia che insegna attualmente: l'entità MATERIA.

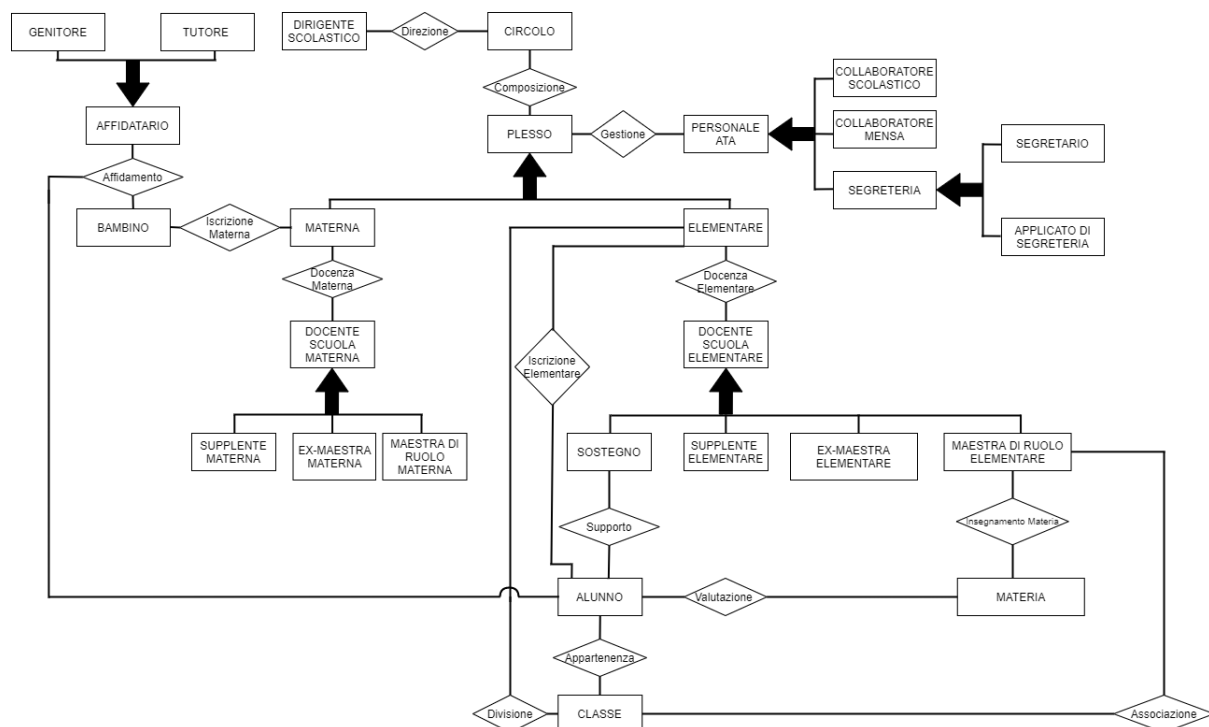
Inoltre la docente di SOSTEGNO dovrà seguire uno o più alunni con la legge 104. Per questo si è inserita la relazione tra le due entità, chiamata Supporto.

Dalle specifiche inoltre si desume che i bambini vengono divisi per età e dunque non hanno bisogno di una relazione con l'entità CLASSE, mentre gli alunni sono divisi per classi in base all'anno che frequentano. Viene dunque definita una relazione Appartenenza tra le entità CLASSE e ALUNNO.

Si è poi definita una relazione Associazione, che associa ad ogni occorrenza di MAESTRA DI RUOLO ELEMENTARE una o più occorrenze dell'entità CLASSE, come definito nelle specifiche della base di dati.

In ultima analisi, è necessario registrare le votazioni conseguite dall'alunno della scuola elementare a determinate materie, per poi ottenere valutazioni periodiche (pagelle) di queste. Per assolvere tale specifica, si è costruita una relazione Valutazione che associa ad ogni occorrenza di MATERIA una o più occorrenze di ALUNNO.

Di seguito, verrà presentato lo schema con le espansioni sopra citate, giunto a ultimazione.

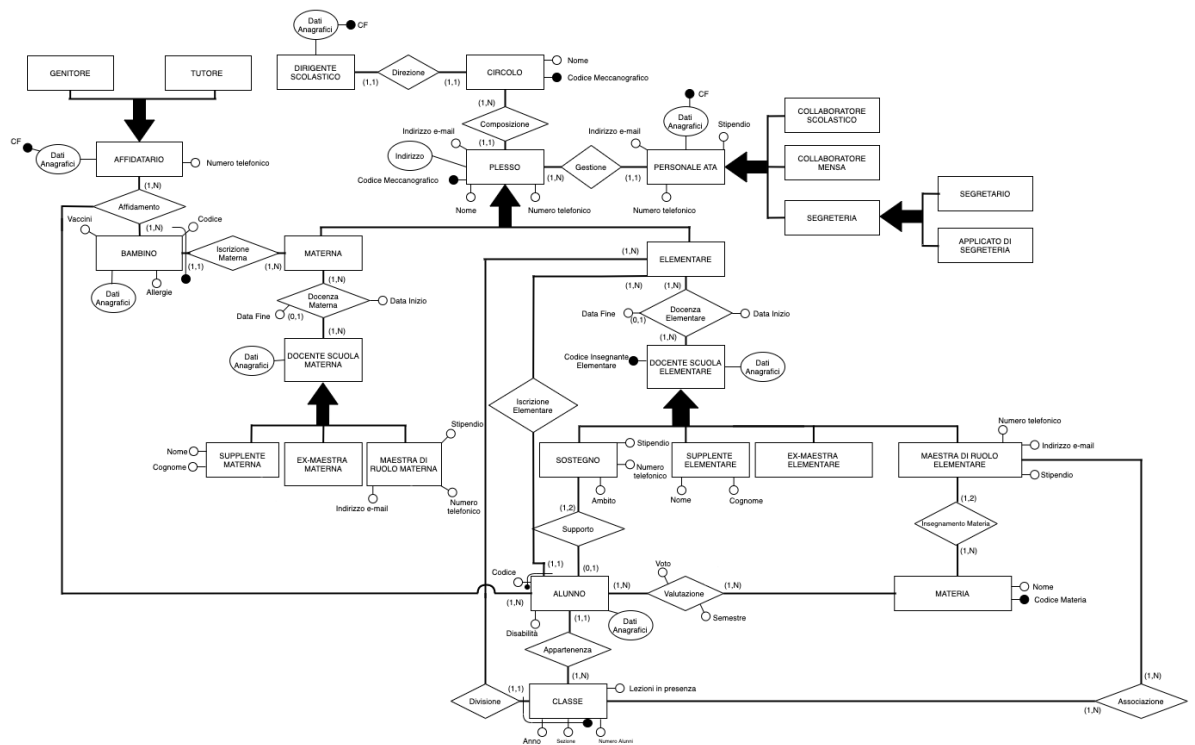


Per terminare lo schema E-R , è necessario introdurre le cardinalità alle relazioni definite e gli attributi delle entità illustrate (compresi gli identificatori).

In particolare, l'attributo composto Dati Anagrafici, comprende a sua volta: Codice Fiscale (CF), Nome, Cognome, Sesso e Data di Nascita. In analogia, l'attributo composto Indirizzo, comprende: Città, Provincia, Via, Numero Civico, CAP.<sup>1</sup>

Di seguito, viene riportato in figura lo schema E-R completo.

<sup>1</sup>Non vengono indicati sullo schema E-R finale per assenza di spazio.



Tuttavia, non è possibile esprimere tutte le specifiche all'interno dello schema E-R. Sarà prevista, dunque, una sezione in cui si tratteranno vincoli non esprimibili (a livello di schema).

## 1.6 Vincoli non esprimibili

*Le informazioni che non possono essere rappresentate con i costrutti del modello E-R vengono elencate a parte e catalogate come vincoli non esprimibili.*

- La valutazione viene fatta in decimi.
- I semestri considerati in un anno scolastico sono due.
- Un insegnante non può esercitare la propria professione contemporaneamente in due scuole diverse nello stesso periodo.



## 1.7 Dizionario dei dati (Entità)

Entità	Identificatore	Attributi	Descrizione
CIRCOLO	Codice Meccanografico	Nome	Circolo scolastico
PLESSO	Codice Meccanografico	Indirizzo, Indirizzo e-mail, Nome, Numero telefonico	Edificio scolastico
DIRIGENTE SCOLASTICO	CF	Dati Anagrafici	Preside del Circolo scolastico
PERSONALE ATA	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Generalizzazione totale ed esclusiva delle tre entità che seguono
COLLABORATORE SCOLASTICO	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Addetto alle pulizie del plesso
COLLABORATORE MENSA	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Addetto al servizio mensa (per le sole scuole materne)
SEGRETERIA	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Generalizzazione totale ed esclusiva delle due entità che seguono
SEGRETARIO	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Impiegato scolastico per lo svolgimento delle mansioni amministrative
APPLICATO DI SEGRETERIA	CF	Dati Anagrafici, Indirizzo e-mail, Numero telefonico, Stipendio	Impiegato scolastico per la gestione interna (back office)
MATERNA	Codice Meccanografico	Nome, Indirizzo, Indirizzo e-mail, Numero telefonico	Edificio scuola materna
ELEMENTARE	Codice Meccanografico	Indirizzo, Indirizzo e-mail, Nome, Numero telefonico	Edificio scuola elementare

DOCENTE SCUOLA MATERNA	Codice Insegnante Materna	Dati Anagrafici	Generalizzazione totale ed esclusiva delle due entità sottostanti
MAESTRA DI RUOLO MATERNA	Codice Insegnante Materna	Indirizzo e-mail, Numero telefonico, Stipendio, Dati Anagrafici	Insegnante di ruolo nella scuola materna
EX-MAESTRA MATERNA	Codice Insegnante Materna	Nome, Cognome	Ex insegnante nella scuola materna
SUPPLENTE MATERNA	Codice Insegnante Materna	Dati Anagrafici, Stipendio, Indirizzo e-mail	Supplente nella scuola materna
DOCENTE SCUOLA ELEMENTARE	Codice Insegnante Elementare	Dati Anagrafici	Generalizzazione totale ed esclusiva delle tre entità che seguono
MAESTRA DI RUOLO ELEMENTARE	Codice Insegnante Elementare	Indirizzo e-mail, Numero telefonico, Dati Anagrafici	Insegnante di ruolo nella scuola elementare
EX-MAESTRA ELEMENTARE	Codice Insegnante Elementare	Nome, Cognome	Ex insegnante nella scuola elementare
SUPPLENTE ELEMENTARE	Codice Insegnante Elementare	Dati Anagrafici	Supplente nella scuola elementare
SOSTEGNO	Codice Insegnante Elementare	Indirizzo e-mail, Numero telefonico, Dati Anagrafici	Insegnante di sostegno per alunni con legge 104
ALUNNO	Codice, Codice Meccanografico (Iscrizione Elementare)	Disabilità	Alunno frequentante la scuola elementare
BAMBINO	Codice, Codice Meccanografico (Iscrizione Materna)	Dati Anagrafici, Vaccini, Allergie	Alunno frequentante la scuola materna
AFFIDATARIO	CF	Dati Anagrafici, Numero telefonico	Generalizzazione totale ed esclusiva delle due entità sottostanti

GENITORE	CF	Dati Anagrafici, Numero telefonico	Affidatario con primo grado di parentela del bambino
TUTORE	CF	Dati Anagrafici, Numero telefonico	Affidatario generico del bambino
MATERIA	Codice Materia	Nome	Insegnamento docenti di scuola elementare
CLASSE	Anno, Sezione, Codice Meccanografico (Divisione)	Lezioni in presenza, Numero Alunni	Classe della scuola elementare

## 1.8 Dizionario dei dati (Relazioni)

Relazione	Entità	Attributi	Descrizione
Composizione	CIRCOLO, PLESSO		Composizione Circolo scolastico
Direzione	CIRCOLO, DIRIGENTE SCOLASTICO		Direzione Circolo scolastico
Gestione	PLESSO, PERSONALE ATA		Gestione amministrativa del plesso
Docenza Elementare	ELEMENTARE, DOCENTE SCUOLA ELEMENTARE		Docenza di scuola elementare
Docenza Materna	MATERNA, DOCENTE SCUOLA MATERNA		Docenza di scuola materna
Iscrizione Elementare	ELEMENTARE, ALUNNO		Iscrizione scuola elementare
Iscrizione Materna	BAMBINO, MATERNA		Iscrizione scuola materna
Affidamento	BAMBINO (ALUNNO), AFFIDATARIO		Affidamento al relativo affidatario del bambino o dell'alunno
Supporto	SOSTEGNO, ALUNNO		Maestra di sostegno che segue un alunno con legge 104
Appartenenza	ALUNNO, CLASSE		Appartenenza di un alunno ad una determinata classe
Valutazione	ALUNNO, MATERIA	Voto, Semestre	Valutazione associata ad un alunno
Insegnamento Materia	MAESTRA DI RUOLO ELEMENTARE, MATERIA		Insegnamento materia della docente di ruolo elementare
Associazione	MAESTRA DI RUOLO ELEMENTARE, CLASSE		Associazione di una maestra ad una o più classi
Divisione	ELEMENTARE, CLASSE		Divisione della scuola elementare in classi

## 2 Progettazione Logica

*È necessario ristrutturare lo schema E-R ottenuto durante la fase di progettazione concettuale, al fine di rendere possibile la traduzione in schema relazionale.*

### 2.1 Tavola dei volumi

Concetto	Tipo	Volume
CIRCOLO	E	1
DIRIGENTE SCOLASTICO	E	1
PLESSO	E	10
MATERNA	E	5
ELEMENTARE	E	5
DOCENTE SCUOLA MATERNA	E	105
MAESTRA DI RUOLO MATERNA	E	35 (5 per ogni scuola)
EX-MAESTRA MATERNA	E	60
SUPPLENTE MATERNA	E	10 (2 per ogni scuola materna)
BAMBINO	E	500 (100 per ogni scuola materna)
DOCENTE SCUOLA ELEMENTARE	E	178
MAESTRA DI RUOLO ELEMENTARE	E	50
EX-MAESTRA ELEMENTARE	E	100
SUPPLENTE ELEMENTARE	E	20
SOSTEGNO	E	8
ALUNNO	E	1.250 (più o meno 250 ogni scuola)
MATERIA	E	11
CLASSE	E	$5(\text{anni}) * 2(\text{sezioni}) * 5(\text{scuole}) = 50$
AFFIDATARIO	E	3.200
GENITORE	E	2.800 (qualche bambino ha fratelli)
TUTORE	E	400
PERSONALE ATA	E	200
COLLABORATORE SCOLASTICO	E	50
COLLABORATORE MENSA	E	100
SEGRETERIA	E	50
SEGRETARIO	E	10
APPLICATO DI SEGRETERIA	E	40
DIREZIONE	R	1
COMPOSIZIONE	R	10
GESTIONE	R	200
DOCENZA ELEMENTARE	R	178
DOCENZA MATERNA	R	105
ISCRIZIONE MATERNA	R	500
ISCRIZIONE ELEMENTARE	R	1.500

SUPPORTO	R	14 (in media una maestra di sostegno segue 1 o 2 alunni)
INSEGNAMENTO MATERIA	R	$50(\text{maestre di ruolo}) * 2 = 100$
ASSOCIAZIONE	R	$80 * 2 * 10 * 5 = 8.000$
APPARTENENZA	R	3.000
DIVISIONE	R	50
VALUTAZIONE	R	$1.250 * 11 * 2 = 33.000$
AFFIDAMENTO	R	$(1.250 + 500) * 2 = 4000$

## 2.2 Tavola delle operazioni

Operazione	Tipo	Frequenza
OP 1: Iscrivere un nuovo alunno.	I	250 ogni anno scolastico
OP 2: Aggiungere un nuovo bambino.	I	100 ogni anno scolastico
OP 3: Aggiungere un docente di scuola primaria.	I	1 ogni anno scolastico
OP 4: Aggiungere un affidatario.	I	2 ogni ciclo scolastico
OP 5: Aggiungere un dipendente del personale ATA.	I	1 ogni anno scolastico
OP 6: Assegnare una nuova supplente di scuola elementare.	I	1 ogni anno scolastico
OP 7: Aggiungere una maestra di ruolo di scuola materna.	I	1 ogni anno scolastico
OP 8: Assegnare una valutazione a un alunno.	I	2 ogni anno scolastico
OP 9: Eliminare gli alunni che hanno concluso il ciclo scolastico.	I	250 ogni anno scolastico
OP 10: Eliminare una maestra.	I	1 ogni 5 anni
OP 11: Sostituire la preside.	I	1 ogni 5 anni
OP 12: Sostituire una docente di ruolo con una supplente.	I	2 ogni semestre
OP 13: Inserire le allergie di un bambino già iscritto alla scuola.	I	1 ogni ciclo scolastico
OP 14: Assegnare una lezione in presenza o online a una classe.	I	1 a settimana
OP 15: Assegnare una materia al docente.	I	2 ogni anno scolastico
OP 16: Aggiornare lo stipendio del personale ATA.	I	1 ogni 5 anni
OP 17: Stampare i voti della pagella di un alunno.	B	2 ogni anno scolastico
OP 18: Stampare i docenti di una classe.	B	1 ogni anno scolastico
OP 19: Stampare gli alunni di una classe.	B	1 ogni anno scolastico
OP 20: Stampare i dati lavorativi del personale ATA del circolo.	B	1 ogni anno scolastico
OP 21: Stampare nome, cognome e codice insegnante di tutti i docenti di un plesso.	B	1 ogni anno scolastico
OP 22: Stampare i codici insegnante di tutti i docenti del circolo.	B	1 ogni anno scolastico
OP 23: Stampare nome, cognome, scuola e classe di uno studente.	B	1.250 volte ogni semestre
OP 24: Stampare gli studenti che non frequentano IRC.	B	1 volta ogni anno scolastico
OP 25: Stampare i dati anagrafici del personale ATA.	B	1 volta ogni anno scolastico
OP 26: Stampare tutte le supplenti che hanno sostituito le docenti presenti e/o passate.	B	1 volta ogni anno scolastico

## 2.3 Analisi delle ridondanze

*Una ridondanza, in uno schema concettuale, consiste nella presenza di un dato che può essere derivato da altri dati mediante una serie di operazioni. L'analisi delle ridondanze è necessaria per comprendere se, una data ridondanza, comporta un vantaggio o uno svantaggio in termini di prestazioni.*

La prima ridondanza da analizzare, risulta quella della relazione Iscrizione Elementare tra ELEMENTARE e ALUNNO, che può essere derivata dalle associazioni Appartenenza e Divisione. Infatti, è possibile notare un ciclo tra le relazioni appena citate.

Per determinare se conviene o meno mantenere questa ridondanza, occorre analizzare e quindi confrontare, gli indici di prestazione nei due casi, con e senza ridondanza. L'indice di prestazione viene calcolato sulla base delle operazioni che coinvolgono relazioni ed entità, oggetto della ridondanza.

In questo caso specifico, l'operazione di cui vanno valutati gli accessi, sono le seguenti.

**OP 1:** Iscrivere un nuovo alunno. (250 ogni anno scolastico, 50 per ogni plesso);

**OP 9:** Eliminare gli alunni che hanno concluso il ciclo scolastico. (250 ogni anno scolastico, 50 per ogni plesso);

**OP 23:** Stampare nome, cognome, scuola e classe di uno studente. (1.250 volte ogni semestre);

**In presenza di ridondanza**, cioè con la relazione Iscrizione Elementare, si hanno le seguenti tavole degli accessi.

Tavola degli Accessi **OP 1:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Iscrizione Elementare	R	1	S
Elementare	E	1	L
Appartenenza	R	1	S
Classe	E	1	L



Tavola degli Accessi **OP 9:** <sup>2</sup>

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Iscrizione Elementare	R	1	S
Appartenenza	R	1	S

Tavola degli Accessi **OP 23:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	L
Iscrizione Elementare	R	1	L
Appartenenza	R	1	L

Considerando il peso di una scrittura (S) pari a 2 letture (L) e moltiplicando i risultati parziali per la frequenza delle operazioni, si ottiene che il costo totale all'anno in termini di accessi in lettura, **in presenza di ridondanza**, è dato dal seguente calcolo.

$$(3S + 2L) * 50 * 5 + 3S * 50 * 5 + 3L * 1.250 * 2 = 8L * 250 + 6L * 250 + 3L * 2500 = 11.000 L \text{ (Letture) all'anno.}$$

Si assume che un'occorrenza della relazione Iscrizione Elementare occupa in memoria 1 byte (per rappresentare un ALUNNO) + 6 byte (per rappresentare una SCUOLA ELEMENTARE).

Si ricorda che un alunno può essere iscritto ad un'unica scuola elementare. La relazione Iscrizione Elementare conta 1.250 occorrenze, per un totale di 8.750 byte.

---

<sup>2</sup>Viene considerato che, nel conteggio degli accessi in una qualunque operazione di modifica, si agisce direttamente modificando il dato senza leggerlo. In tutti gli altri casi, si considera anche l'accesso in lettura del dato.

In assenza di ridondanza, si hanno le seguenti tavole degli accessi.

Tavola degli Accessi **OP 1:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Appartenenza	R	1	S
Classe	E	1	L

Tavola degli Accessi **OP 9:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Appartenenza	R	1	S

Tavola degli Accessi **OP 23:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	L
Appartenenza	R	1	L
Classe	E	1	L
Divisione	R	1	L

Di seguito, segue il numero di accessi in lettura annuali, nel caso di **assenza di ridondanza**.

$$\begin{aligned}
 &(2S + 1L) \cdot 50 \cdot 5 + (2S) \cdot 50 \cdot 5 + 4L \cdot 1.250 \cdot 2 = \\
 &= 5L \cdot 250 + 4L \cdot 250 + 4L \cdot 2500 = 12.250 \text{ L (Lettture) all'anno.}
 \end{aligned}$$

Si ottengono 1.250 accessi in più rispetto al caso di dato ridondante presente, contro un risparmio di 8.750 byte ( < 9 kbyte).

Si conclude di mantenere il dato ridondante. In tal caso, dunque, lo schema E-R rimane invariato.

L'altra ridondanza da considerare, riguarda l'attributo Numero Alunni dell'entità CLASSE. Questo, infatti, potrebbe essere ricavato indirettamente contando le occorrenze della relazione Appartenenza.

In questo caso specifico, l'operazione di cui vanno valutati gli accessi è la seguente.

**OP 1:** Iscrivere un nuovo alunno. (250 ogni anno scolastico per ogni plesso);

Di seguito, si considera la tavola degli accessi **in presenza di ridondanza**.

Tavola degli Accessi **OP 1:**

Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Appartenenza	R	1	S
Classe	E	1	L
Classe	E	1	S

**In presenza di ridondanza**, si ha il seguente numero di accessi in lettura annuale.

$$(3S + 1L) * 50 * 5 = 7L * 250 = 1.750 \text{ L (Letture) all'anno.}$$

Si assume che, per memorizzare il numero di alunni di una classe, si utilizza 1 byte (più che sufficiente allo scopo).

L'attributo Numero Alunni conta 50 occorrenze (perchè in totale le classi sono 50), per un totale di 50 byte.

Di seguito, si prende in considerazione la tavola degli accessi in **assenza di ridondanza**.

Tavola degli Accessi **OP 1:**

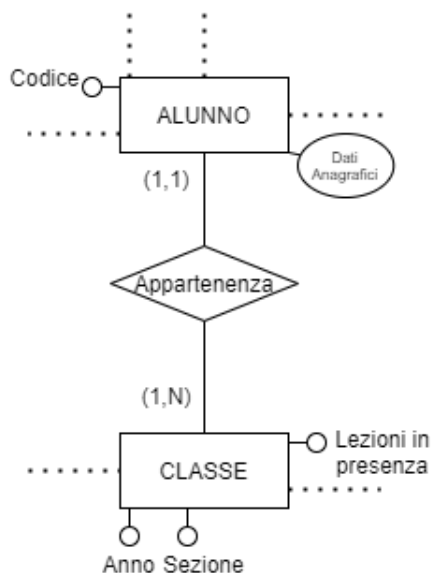
Concetto	Costrutto	Accessi	Tipo
Alunno	E	1	S
Appartenenza	R	1	S
Classe	E	1	L

**In assenza di ridondanza**, si ottiene il seguente numero di accessi in lettura annuali.

$$(2S + 1L) * 50 * 5 = 5L * 250 = 1.250L \text{ (Letture) all'anno.}$$

Senza ridondanza si ottiene un numero minore di letture (500 letture in meno) e si risparmiano 50 byte, si sceglie quindi di eliminarla.

In questo caso, si elimina l'attributo Numero Alunni dall'entità CLASSE, come indicato nella figura che segue.



In ultimo, si analizza la ridondanza riguardante la relazione Docenza Elementare, tra le entità ELEMENTARE e DOCENTE SCUOLA ELEMENTARE. Anch'essa può essere derivata dalle relazioni Associazione e Divisione, poiché si è in presenza di un ciclo.

In tal caso, le operazioni da considerare, risultano le seguenti.

**OP 3:** Aggiungere un docente di scuola primaria. (1 volta ogni anno scolastico);

**OP 6:** Assegnare una nuova supplente di scuola elementare. (1 volta ogni anno scolastico);

**OP 10:** Eliminare una maestra. (1 volta ogni cinque anni);

**OP 21:** Stampare nome, cognome e codice insegnante di tutti i docenti di un plesso. (1 volta ogni anno per ogni circolo);

Tuttavia l'**OP 10**, a causa della poca frequenza con la quale avviene, è trascurabile. Non verranno considerate le tavole degli accessi in questo caso.

Di seguito si prendono in considerazione le tavole degli accessi **in presenza di ridondanza** delle operazioni rimanenti.

Tavola degli Accessi **OP 3:**

Concetto	Costrutto	Accessi	Tipo
Docenza Elementare	R	1	S
Docente Scuola Elementare	E	1	S
Elementare	E	1	L
Insegnamento Materia	R	1	S
Associazione	R	1	S
Materia	E	1	L
Classe	R	1	L

Tavola degli Accessi **OP 6:**

Concetto	Costrutto	Accessi	Tipo
Supplente Elementare	E	1	S
Docenza Elementare	R	1	S
Elementare	E	1	L

Tavola degli Accessi **OP 21:**<sup>3</sup>

Concetto	Costrutto	Accessi	Tipo
Elementare	E	1	L
Docente Scuola Elementare	E	35	L
Docenza Elementare	R	35	L
Materna	E	1	L
Docente Scuola Materna	E	21	L
Docenza Materna	R	21	L

---

<sup>3</sup>Non tutte le supplenti lavorano.

Si ottiene, **in presenza di ridondanza**, il seguente numero di accessi in lettura annuali.

$$(4S + 3L)*1 + (2S + 1L)*1 + 114L*1*56 = 11L*2 + 5L*2 + 114L*56 = 6.416 L \text{ (Letture) all'anno.}$$

Si assuma che per memorizzare un'occorrenza dell'entità DOCENTE SCUOLA ELEMENTARE e si usino 4 byte, mentre un unico byte per un'occorrenza dell'entità ELEMENTARE.

Poiché tutte le insegnanti lavorano in un unico plesso, la relazione di Docenza Scuola Elementare ha 178 occorrenze e occupa uno spazio di 890 byte.

**In assenza di ridondanza**, si hanno le seguenti tavole degli accessi.

Tavola degli Accessi **OP 3:**

Concetto	Costrutto	Accessi	Tipo
Classe	E	1	L
Associazione	R	1	S
Insegnamento Materia	R	1	S
Docente Scuola Elementare	E	1	S
Materia	E	1	L

Tavola degli Accessi **OP 6:**

Concetto	Costrutto	Accessi	Tipo
Supplente Elementare	E	1	S

Tavola degli Accessi **OP 21:**<sup>4</sup>

Concetto	Costrutto	Accessi	Tipo
Sostegno	E	2	L
Maestra di Ruolo Elementare	E	10	L
Classe	E	10	L

---

<sup>4</sup>Con lo schema E-R realizzato non c'è modo di collegare la supplente e con la scuola elementare (si analizza questo nelle fasi a seguire)

Supporto	R	2	L
Alunno	E	2	L
Iscrizione Elementare	R	2	L
Elementare	E	1	L
Associazione	R	10	L
Divisione	R	10	L
Materna	E	1	L
Docente Scuola Materna	E	21	L
Docenza Materna	R	21	L

Si ottiene, **in assenza di ridondanza**, il seguente numero di accessi in lettura annuali.

$(3S + 2L)*2 + (1S)*2 + 92L*56*1 = 8L*2 + 2L*2 + 92L*56 = 5.172 \text{ L}$   
(Letture) all'anno.

In assenza di ridondanza si ottengono 1.244 accessi in meno rispetto al caso di dato ridondante presente, con un risparmio di 890 byte ( < 1 kbyte).

Si decide di mantenere il dato ridondante, in quanto si risolverà il problema del dato ridondante nella fase del partizionamento di relazione.

## 2.4 Eliminazione delle generalizzazioni

*Le generalizzazioni non sono direttamente rappresentabili nei DBMS. Di seguito si trasformano tali costrutti utilizzando esclusivamente entità e relazioni.*

Si procede con la seguente strategia.

Si analizzeranno prima le generalizzazioni esterne, quelle più semplici, perché le entità figlie sono sprovviste di collegamento di tipo relazione, per poi procedere con le entità centrali nello schema.

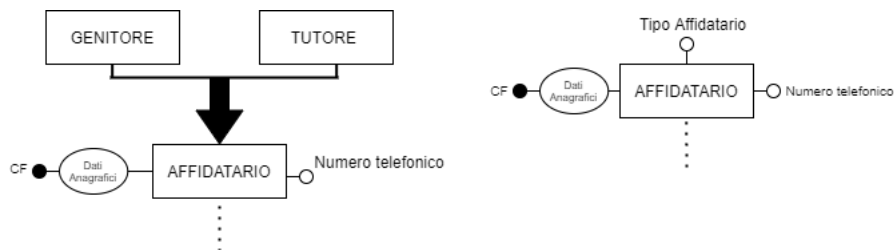
Per rappresentare una generalizzazione mediante entità e associazioni, si hanno essenzialmente tre alternative possibili. Di seguito, vengono elencate solo quelle utilizzate ( la (1) e la (3) ).

### (1) Accorpamento delle Figlie della Generalizzazione nel Genitore

Iniziamo l'analisi delle generalizzazioni dell'entità genitore AFFIDATARIO. Poiché nelle operazioni, non è di interesse distinguere tra le due entità figlie (GENITORE e TUTORE) si decide di accorparle nel genitore.

Si aggiunge un unico attributo Tipo Affidatario che separa i concetti considerati: Genitore e Tutore.

La figura seguente mostra lo schema prima e dopo l'eliminazione.

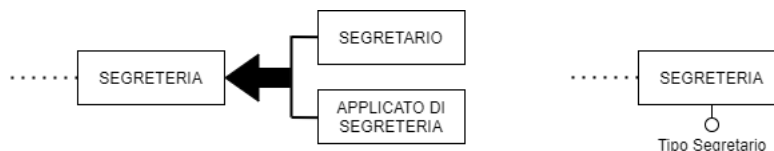


Continuando poi con l'analisi delle generalizzazioni, si considera l'entità genitore SEGRETERIA al primo livello di gerarchia. Anche in questo caso, nelle operazioni, non è di interesse distinguere tra le due entità figlie (SEGRETERARIO e APPLICATO DI SEGRETERIA). Si decide, dunque, di accorparle nel genitore.

Analogamente, si inserisce un unico attributo Tipo Segretario che separa i due concetti appena citati: Segretario e Applicato Segreteria.



La figura seguente mostra lo schema prima e dopo l'eliminazione.



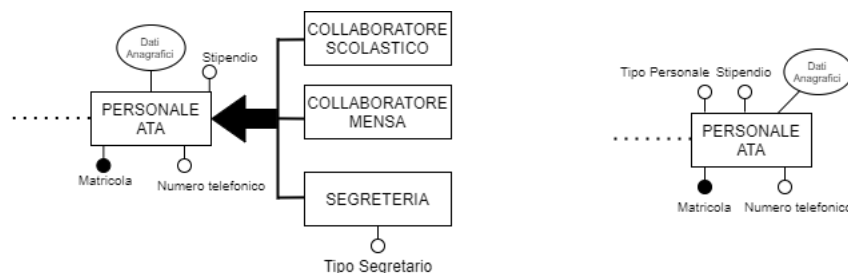
Si considera ora, il livello superiore di gerarchia, che considera l'entità genitore PERSONALE ATA.

Valgono, in questo caso, le stesse considerazioni fatte in precedenza. Si procede accorpando le entità figlie COLLABORATORE SCOLASTICO, COLLABORATORE MENSA e SEGRETERIA nell'entità genitore PERSONALE ATA. Si introduce, a tal fine, un nuovo attributo Tipo Personale che distingue i possibili componenti dell'entità genitore.

Per evitare ulteriori problemi, si è pensato di considerare direttamente i componenti della segreteria, Segretario e Applicato di Segreteria all'interno di Tipo Personale, piuttosto che Tipo Segretario come definito in precedenza.

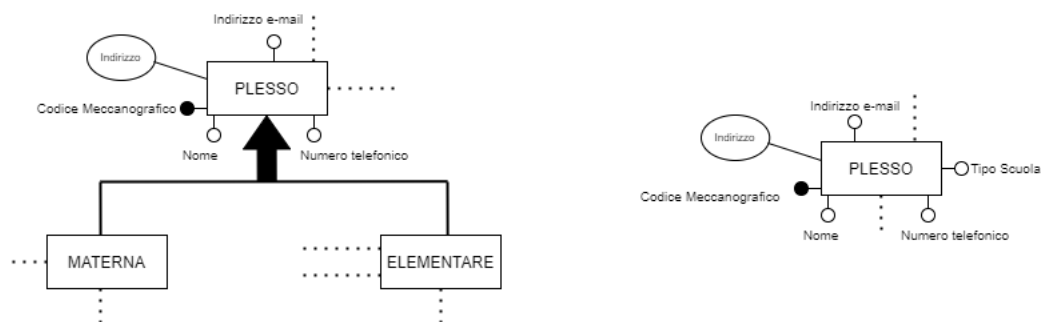
In questo caso non sono presenti operazioni che distinguono i vari componenti del personale.

Di seguito, si riporta in figura, lo schema prima e dopo l'eliminazione.



Poiché non si hanno operazioni separate per le entità MATERNA ed ELEMENTARE, figlie dell'entità genitore PLESSO, si elimina la generalizzazione introducendo un nuovo attributo Tipo Scuola, per separare Materna ed Elementare. Si procede in modo analogo a quanto fatto sopra.

La figura seguente mostra lo schema prima e dopo l'eliminazione.



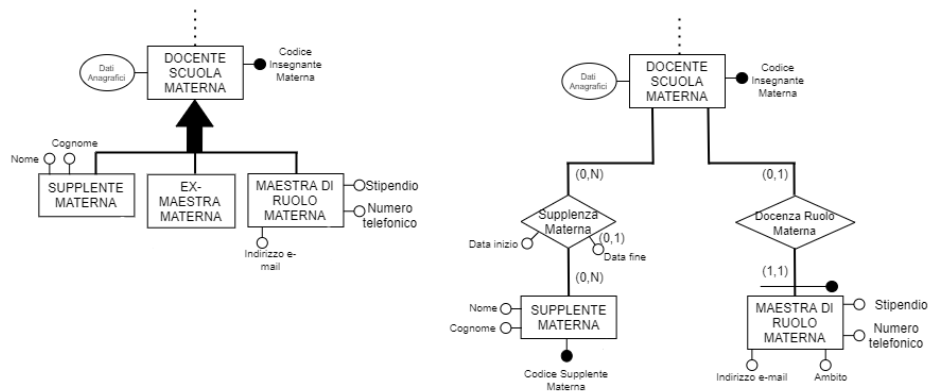
### (3) Sostituzione della Generalizzazione con Relazioni

Si va ora a considerare la generalizzazione tra **DOCENTE SCUOLA MATERNA** (entità genitore), e le tre entità figlie **MAESTRA DI RUOLO MATERNA**, **EX-MAESTRA MATERNA**, **SUPPLENTE MATERNA**.

Si introducono due relazioni distinte, **Supplenza Materna** e **Docenza Ruolo Materna**, che legano le due figlie al genitore. Questo cambiamento risulta una necessità, a causa delle operazioni che distinguono i due concetti. Inoltre, si introduce un identificatore esterno per **MAESTRA DI RUOLO MATERNA**; in tal modo, è possibile identificare la maestra di ruolo con il relativo codice insegnante materna, attributo dell'entità genitore.

L'entità **SUPPLENTE MATERNA** viene ora identificata con l'apposito codice supplente, diverso da quello insegnante. **EX-MAESTRA MATERNA** viene invece accorpata, secondo un'eliminazione "ibrida", nella parte superiore dello schema E-R. Qui, si andranno a definire due nuove relazioni che descriveranno il concetto accorpato con maggior dettaglio.

Di seguito, in figura, si mostra lo schema prima e dopo l'eliminazione.



In ultima analisi, si considera la generalizzazione tra **DOCENTE SCUOLA ELEMENTARE** (entità genitore), e le quattro entità figlie **MAESTRA DI RUOLO ELEMENTARE**, **EX-MAESTRA ELEMENTARE**, **SOSTEGNO** e **SUPPLENTE ELEMENTARE**.

Si introducono tre relazioni distinte: **Docenza Ruolo Elementare**, **Supplenza Elementare** e **Docenza Sostegno**. Queste, legano le figlie al genitore. Il cambiamento apportato, risulta una necessità, a causa delle operazioni che distinguono i tre concetti.

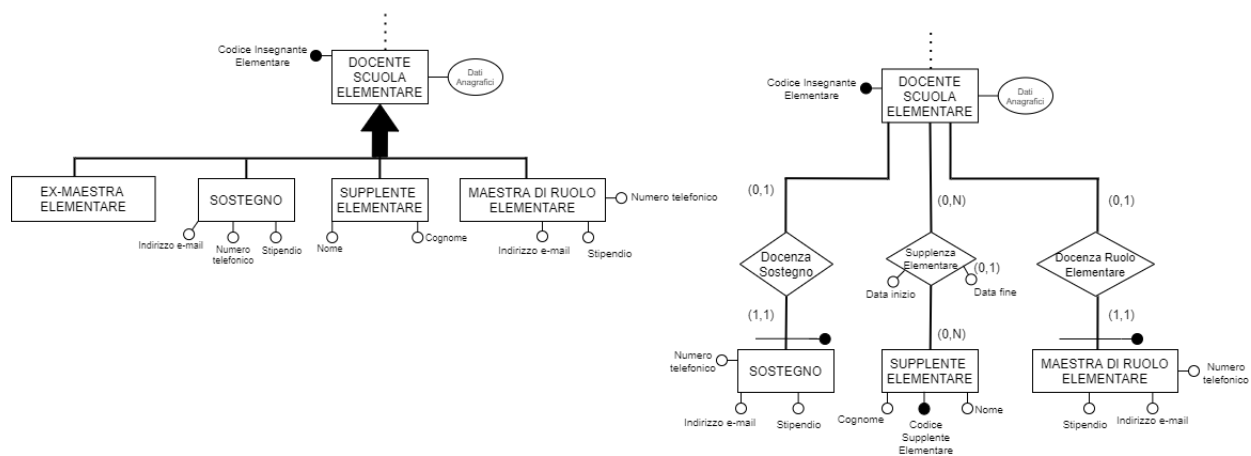
Inoltre, si introduce un identificatore esterno per **MAESTRA DI RUOLO ELEMENTARE** e per **SOSTEGNO**; in tal modo, è possibile identificare la maestra di ruolo e di sostegno (anch'essa considerata come "di ruolo") con il relativo codice insegnante elementare, attributo dell'entità genitore.

L'entità **SUPPLENTE ELEMENTARE** viene identificata ora con l'apposito codice supplente, diverso da quello insegnante. **EX-MAESTRA ELEMENTARE** viene invece accorpata, secondo un'eliminazione "ibrida", nella parte superiore dello schema E-R. Qui, si andranno a definire due nuove relazioni che descriveranno il concetto accorpato con maggior dettaglio.

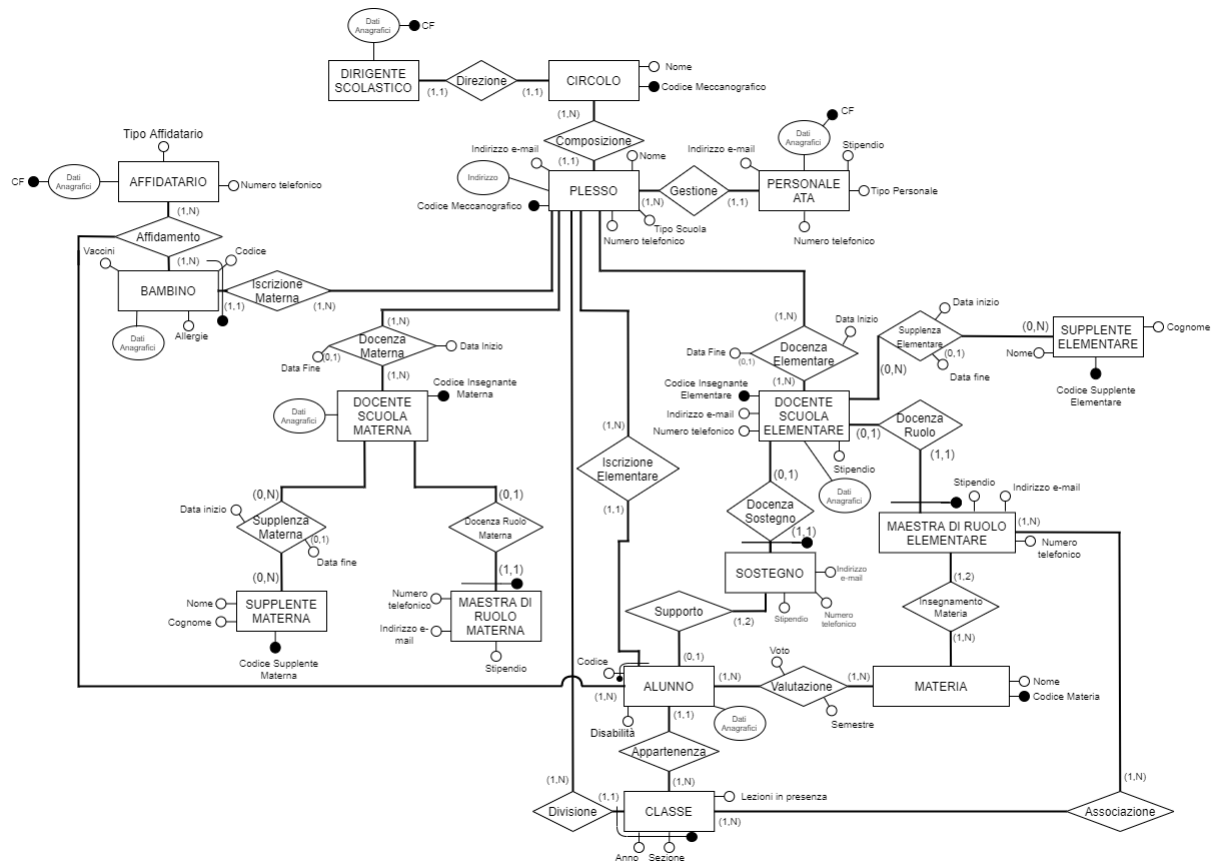
Di seguito, in figura, si mostra lo schema prima e dopo l'eliminazione.

Si è deciso di operare in questo modo per le ultime due generalizzazioni, per evidenziare la differenza tra **DOCENTE SCUOLA ELEMENTARE** e **MAESTRA DI RUOLO ELEMENTARE**.

Con la prima entità, si vogliono indicare tutte le docenti che hanno insegnato almeno una volta nel Circolo di Foligno (si considerano sia le docenti in pensione, sia le supplenti che hanno fatto almeno una sostituzione), mentre con l'entità **MAESTRA DI RUOLO ELEMENTARE**, si vogliono indicare le docenti che attualmente lavorano in una scuola. Un'analoga considerazione si è fatta per le entità **DOCENTE SCUOLA MATERNA** e **MAESTRA DI RUOLO MATERNA**.



Di seguito, si riporta in figura, lo schema E-R ristrutturato dopo le fasi di analisi delle ridondanze ed eliminazione delle generalizzazioni.

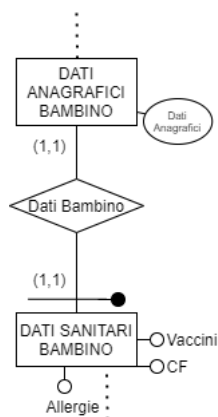


## 2.5 Partizionamento di Entità e Relazioni

*Entità e associazioni in uno schema E-R possono essere partizionati per garantire una maggior efficienza delle operazioni: gli accessi si riducono separando attributi di uno stesso concetto che vengono acceduti da operazioni diverse.*

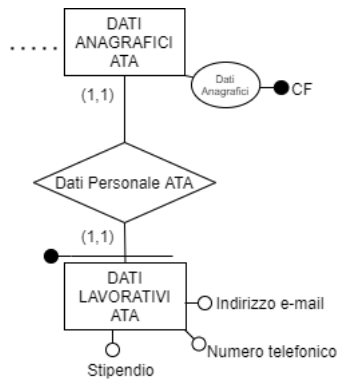
Si inizia analizzando l'entità BAMBINO, per cui le operazioni previste, sono diversificate in base ai suoi dati sanitari (allergie, vaccini) e ai suoi dati anagrafici, tra cui il codice fiscale con il quale lo si identifica.

Per questo, si procede con un partizionamento verticale dell'entità BAMBINO, considerando due nuove entità distinte (DATI ANAGRAFICI BAMBINO e DATI SANITARI BAMBINO), collegate da una relazione uno a uno (Dati Bambino).



Analogo ragionamento viene fatto per l'entità PERSONALE ATA. Le operazioni considerate, interessano separatamente i dati lavorativi del personale e i loro dati anagrafici, tra cui l'attributo CF con il quale identifichiamo l'entità.

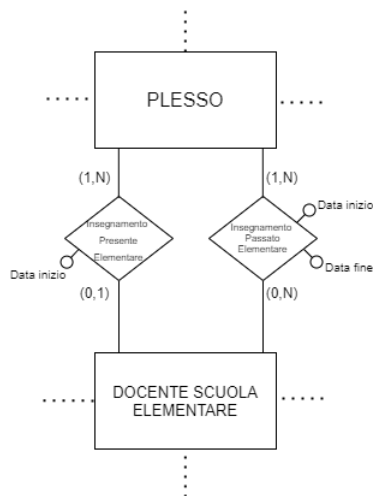
Si procede, dunque, con un partizionamento verticale dell'entità PERSONALE ATA, considerando due nuove entità distinte (DATI ANAGRAFICI ATA e DATI LAVORATIVI ATA), collegate da una relazione uno a uno (Dati Personale ATA).



Si considera ora, il partizionamento riguardante la relazione Docenza Elementare. In tal caso, è possibile concentrare l'interesse sull'insegnamento presente e passato della docente di scuola elementare.

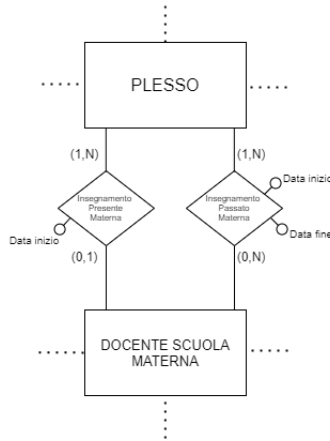
Si partiziona la relazione in due associazioni: Insegnamento Passato Elementare e Insegnamento Presente Elementare, tra le due entità PLESSO e DOCENTE SCUOLA ELEMENTARE. Inoltre, vengono aggiunti degli attributi a queste relazioni: la data d'inizio del servizio presso la scuola elementare e la data di fine (solo nel caso di docenza passata).

Di seguito, in figura, si illustra la decomposizione appena esplicitata.



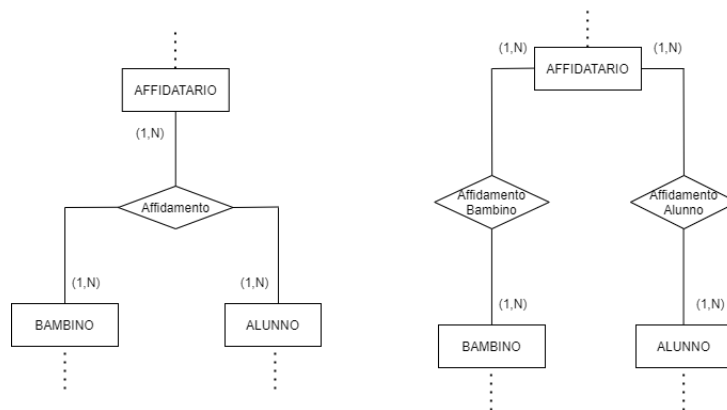
Analogamente, si considera lo stesso partizionamento per la relazione Docenza Materna.

Di seguito, in figura si mostra la decomposizione appena esplicitata.



In ultimo, si considera il partizionamento della relazione Affidamento in due relazioni Affidamento Bambino e Affidamento Alunno. La necessità di effettuarlo è data dal fatto che si avrebbero troppi valori nulli nei casi in cui l'affidatario sia legato al solo bambino o al solo alunno. Inoltre, le operazioni nei due casi differenti, sono separate.

Di seguito, si mostra in figura il partizionamento della relazione.





## 2.6 Accorpamento di Entità e Relazioni

*L'accorpamento rappresenta l'operazione inversa del partizionamento. Anche quest'ultimo, è volto al raggiungimento di una maggior efficienza nell'esecuzione delle operazioni.*

Lo schema non presenta possibili relazioni o entità da accorpare.

## 2.7 Scelta degli identificatori primari

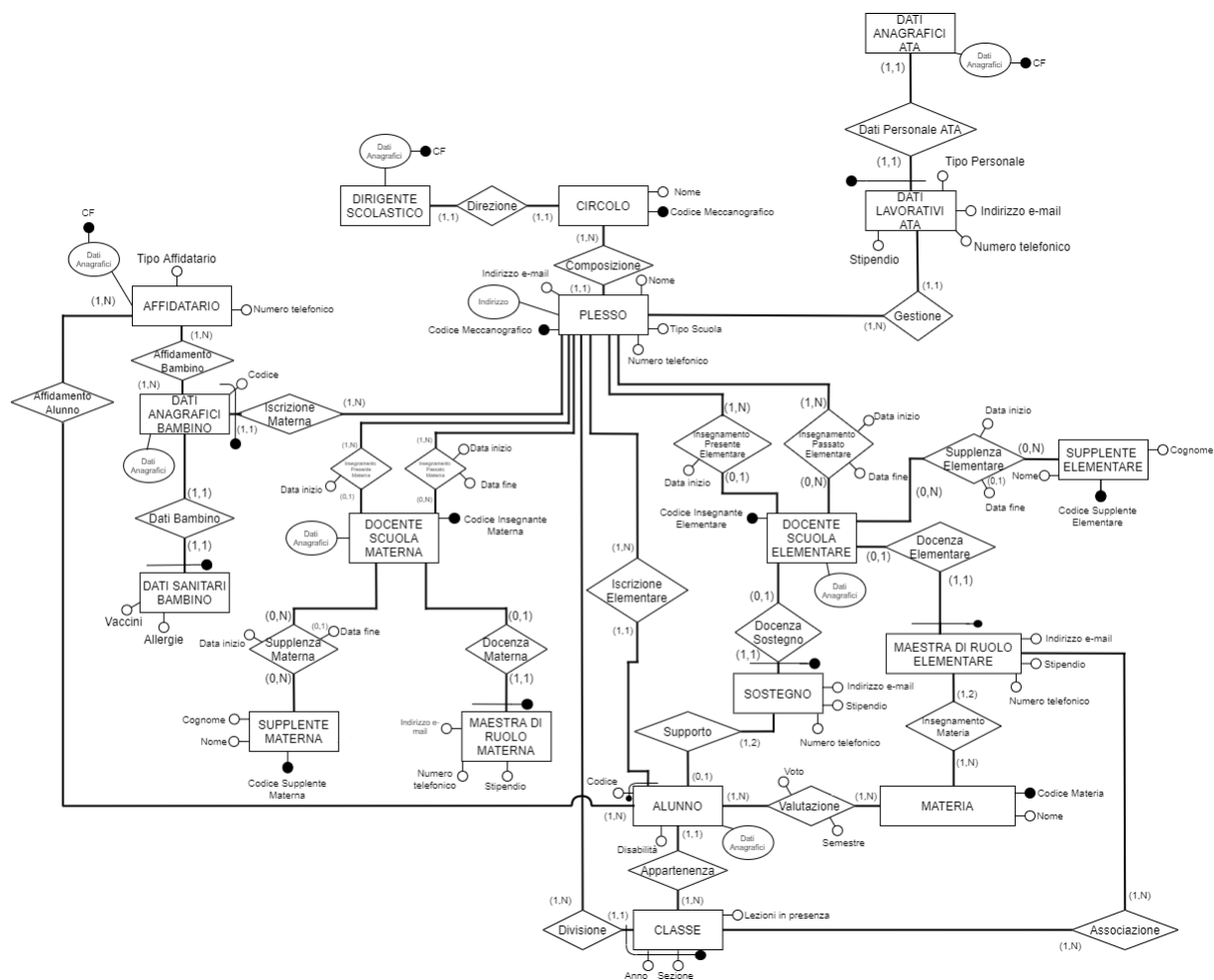
*La scelta degli identificatori principali è molto importante nelle traduzioni verso il modello relazionale, poiché in tale modello, le chiavi vengono utilizzate per stabilire legami tra dati in relazioni diverse.*

Osservando lo schema E-R ristrutturato finora si può verificare che molte entità hanno più di una chiave primaria.

Per fare in modo di distinguere le varie entità tra loro, si decide di scegliere un unico identificatore primario diverso per ognuna di esse, come viene illustrato nella seguente tabella.

ENTITÀ	IDENTIFICATORE PRIMARIO
DIRIGENTE SCOLASTICO	Codice Fiscale (CF)
CIRCOLO	Codice meccanografico (Circolo)
PLESSO	Codice meccanografico (Plesso)
DOCENTE SCUOLA MATERNA	Codice Insegnante Materna
SUPPLENTE MATERNA	Codice Supplente Materna
MAESTRA DI RUOLO MATERNA	Codice Insegnante Materna
SUPPLENTE ELEMENTARE	Codice Supplente Elementare
DOCENTE SCUOLA ELEMENTARE	Codice Insegnante Elementare
MAESTRA DI RUOLO ELEMENTARE	Codice Insegnante Elementare
SOSTEGNO	Codice Insegnante Elementare
MATERIA	Codice Materia
ALUNNO	Codice; Codice Meccanografico Plesso (Iscrizione Elementare)
CLASSE	Anno, Sezione, Scuola (Divisione)
AFFIDATARIO	Codice Fiscale (CF)
DATI SANITARI BAMBINO	Codice; Codice Meccanografico Plesso (Dati Bambino)
DATI ANAGRAFICI BAMBINO	Codice; Codice Meccanografico Plesso (Iscrizione Materna)
DATI ANAGRAFICI ATA	Codice Fiscale (CF)
DATI LAVORATIVI ATA	CF (Dati Personale ATA)

Di seguito si mostra in figura lo schema E-R completamente ristrutturato.



## 2.8 Modello relazionale

*Si costruisce uno schema logico equivalente allo schema E-R , in grado di rappresentare le medesime informazioni. Tale schema, chiamato modello relazione, costituisce il risultato della progettazione logica.*

### 2.8.1 Traduzione di Entità

- DIRIGENTE\_SCOLASTICO (CF, Nome, Cognome, Sesso, DataDiNascita)
- CIRCOLO (CodiceMeccanograficoCircolo, Nome, DirigenteScolastico)  
Vincolo di integrità referenziale tra l'attributo DirigenteScolastico della relazione CIRCOLO e la relazione DIRIGENTE\_SCOLASTICO.
- PLESSO (CodiceMeccanograficoPlesso, Nome, Città, Provincia, Via, NumeroCivico, CAP, IndirizzoMail, TipoScuola, NumeroTelefonico, ComposizioneCircolo)  
Vincolo di integrità referenziale tra l'attributo ComposizioneCircolo della relazione PLESSO e la relazione CIRCOLO.
- DOCENTE\_SCUOLA\_MATERNA (CodiceInsegnanteMaterna, CF, Nome, Cognome, Sesso, DataDiNascita)
- SUPLENTE\_MATERNA (CodiceSupplenteMaterna, Nome, Cognome)
- MAESTRA\_DI\_RUOLO\_MATERNA (CodiceInsegnanteMaterna, CF, IndirizzoMail, NumeroTelefonico, Stipendio)  
Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteMaterna della relazione MAESTRA\_DI\_RUOLO\_MATERNA e la relazione DOCENTE\_SCUOLA\_MATERNA.
- DOCENTE\_SCUOLA\_ELEMENTARE (CodiceInsegnanteElementare, CF, Nome, Cognome, Sesso, DataDiNascita)
- SUPLENTE\_ELEMENTARE (CodiceSupplenteElementare, Nome, Cognome)
- MAESTRA\_DI\_RUOLO\_ELEMENTARE (CodiceInsegnanteElementare, IndirizzoMail, NumeroTelefonico, Stipendio)  
Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteElementare della relazione MAESTRA\_DI\_RUOLO\_ELEMENTARE e la relazione DOCENTE\_SCUOLA\_ELEMENTARE.
- SOSTEGNO (CodiceInsegnanteElementare, IndirizzoMail, NumeroTelefonico, Stipendio)

Vincolo di integrità referenziale tra l'attributo `CodiceInsegnanteElementare` della relazione `SOSTEGNO` e la relazione `DOCENTE_SCUOLA_ELEMENTARE`

- `ALUNNO` (`CodiceAlunno`, `CodiceMeccanograficoPlesso`, `CF`, `Nome`, `Cognome`, `Sesso`, `DataDiNascita`, `Disabilità`)

Vincolo di integrità referenziale l'attributo `CodiceMeccanograficoPlesso` di `ALUNNO` e la relazione `PLESSO`.

- `CLASSE` (`CodiceMeccanograficoPlesso`, `Anno`, `Sezione`, `LezioneInPresenza`)

Vincolo di integrità referenziale l'attributo `CodiceMeccanograficoPlesso` di `CLASSE` e la relazione `PLESSO`.

- `MATERIA` (`CodiceMateria`, `Nome`)

- `AFFIDATARIO` (`CF`, `Nome`, `Cognome`, `Sesso`, `DataDiNascita`, `TipoAffidatario`, `NumeroTelefonico`)

- `DATI_ANAGRAFICI_BAMBINO` (`CodiceBambino`, `CodiceMeccanograficoPlesso`, `CF`, `Nome`, `Cognome`, `Sesso`, `DataDiNascita`)

Vincolo di integrità referenziale tra `CodiceMeccanograficoPlesso` e la relazione `PLESSO`.

- `DATI_SANITARI_BAMBINO` (`DatiAnagraficiBambino`, `Vaccini`, `Allergie`)

Vincolo di integrità referenziale tra l'attributo `DatiAnagraficiBambino` della relazione `DATI_SANITARI_BAMBINO` e la relazione `DATI_ANAGRAFICI_BAMBINO`.

- `DATI_ANAGRAFICI_ATA` (`CF`, `Nome`, `Cognome`, `Sesso`, `DataDiNascita`)

- `DATI_LAVORATIVI_ATA` (`DatiAnagraficiATA`, `IndirizzoMail`, `NumeroTelefonico`, `Stipendio`, `TipoPersonale`, `Plesso`)

Vincolo di integrità referenziale tra l'attributo `DatiAnagraficiATA` appartenente a `DATI_LAVORATIVI_ATA` e la relazione `DATI_ANAGRAFICI_ATA`, tra l'attributo `Plesso` e la relazione `PLESSO`.

### 2.8.2 Traduzione di Relazioni

- AFFIDAMENTO\_BAMBINO (CodiceAffidatario,  
CodiceBambino, CodiceMeccanograficoPlesso)

Vincolo di integrità referenziale tra l'attributo CodiceAffidatario della relazione AFFIDAMENTO\_BAMBINO e la relazione AFFIDATARIO, tra l'attributo CodiceBambino e la relazione BAMBINO, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.

- AFFIDAMENTO\_ALUNNO (CodiceAffidatario, CodiceAlunno,  
CodiceMeccanograficoPlesso)

Vincolo di integrità referenziale tra l'attributo CodiceAffidatario della relazione AFFIDAMENTO\_ALUNNO e la relazione AFFIDATARIO, tra l'attributo CodiceAlunno e la relazione ALUNNO, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.

- APPARTENENZA (CodiceAlunno, CodiceMeccanograficoPlesso,  
Anno, Sezione)

Vincolo di integrità referenziale tra l'attributo CodiceAlunno della relazione APPARTENENZA e la relazione ALUNNO, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO, tra gli attributi Anno e Sezione della relazione CLASSE.

- ASSOCIAZIONE (CodiceInsegnanteElementare, CodiceMeccanograficoPlesso,  
Anno, Sezione)

Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteElementare della relazione ASSOCIAZIONE e la relazione DOCENTE\_SCUOLA\_ELEMENTARE, tra l'attributo CodiceMeccanograficoPlesso della relazione PLESSO, tra gli attributi Anno e Sezione della relazione CLASSE.

- INSEGNAMENTO\_PASSATO\_ELEMENTARE (CodiceInsegnanteElementare,  
CodiceMeccanograficoPlesso, DataInizio, DataFine)

Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteElementare della relazione INSEGNAMENTO\_PASSATO\_ELEMENTARE e la relazione DOCENTE\_SCUOLA\_ELEMENTARE, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.

- INSEGNAMENTO\_PRESENTE\_ELEMENTARE (CodiceInsegnanteElementare,  
CodiceMeccanograficoPlesso, DataInizio)

Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteElementare della relazione INSEGNAMENTO\_PRESENTE\_ELEMENTARE e la relazione DOCENTE\_SCUOLA\_ELEMENTARE, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.

- INSEGNAMENTO\_PASSATO\_MATERNA (CodiceInsegnanteMaterna, CodiceMeccanograficoPlesso, DataInizio, DataFine)  
Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteMaterna della relazione INSEGNAMENTO\_PASSATO\_MATERNA e la relazione DOCENTE\_SCUOLA\_MATERNA, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.
- INSEGNAMENTO\_PRESENTE\_MATERNA (CodiceInsegnanteMaterna, CodiceMeccanograficoPlesso, DataInizio)  
Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteMaterna della relazione INSEGNAMENTO\_PRESENTE\_MATERNA e la relazione DOCENTE\_SCUOLA\_MATERNA, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.
- SUPPLENZA\_ELEMENTARE (CodiceSupplenteElementare, CodiceInsegnanteElementare, DataInizio, DataFine\*)  
Vincolo di integrità referenziale tra l'attributo CodiceSupplenteElementare della relazione SUPPLENZA\_ELEMENTARE e la relazione SUPPLENTE\_ELEMENTARE, tra l'attributo CodiceInsegnanteElementare e la relazione DOCENTE\_SCUOLA\_ELEMENTARE.
- SUPPLENZA\_MATERNA (CodiceSupplenteMaterna, CodiceInsegnanteMaterna, DataInizio, DataFine\*)  
Vincolo di integrità referenziale tra l'attributo CodiceSupplenteMaterna della relazione SUPPLENZA\_MATERNA e la relazione SUPPLENTE\_MATERNA, tra l'attributo CodiceInsegnanteMaterna e la relazione DOCENTE\_SCUOLA\_MATERNA.
- INSEGNAMENTO\_MATERIA (CodiceInsegnanteElementare, Materia)  
Vincolo di integrità referenziale tra l'attributo CodiceInsegnanteElementare della relazione INSEGNAMENTO\_MATERIA e la relazione DOCENTE\_SCUOLA\_ELEMENTARE, tra l'attributo Materia e la relazione MATERIA.
- INSEGNAMENTO\_SOSTEGNO (Sostegno, CodiceAlunno, CodiceMeccanograficoPlesso)  
Vincolo di integrità referenziale tra l'attributo Sostegno della relazione INSEGNAMENTO\_SOSTEGNO e la relazione SOSTEGNO, tra l'attributo CodiceAlunno e la relazione ALUNNO, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO.
- VALUTAZIONE (Alunno, CodiceMeccanograficoPlesso, Materia, Voto, Semestre)  
Vincolo di integrità referenziale tra l'attributo Alunno della relazione VALUTAZIONE e la relazione ALUNNO, tra l'attributo CodiceMeccanograficoPlesso e la relazione PLESSO, tra l'attributo Materia e la relazione MATERIA.

## 3 Implementazione delle operazioni

### 3.1 Creazione delle tabelle

```
CREATE DOMAIN sesso AS char
    CHECK (value = 'M' or value = 'F')

CREATE TABLE DIRIGENTE_SCOLASTICO (
    CF char(16) PRIMARY KEY,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL,
    Sesso sesso NOT NULL,
    DataDiNascita date NOT NULL
)

CREATE TABLE CIRCOLO (
    CodiceMeccanograficoCircolo char(10) PRIMARY KEY,
    Nome varchar(20) NOT NULL,
    DirigenteScolastico char(16) NOT NULL,
    FOREIGN KEY (DirigenteScolastico)
    REFERENCES DIRIGENTE_SCOLASTICO (CF)
    ON UPDATE CASCADE
    ON DELETE CASCADE
)

CREATE TABLE PLESSO (
    CodiceMeccanograficoPlesso char(10) PRIMARY KEY,
    Nome varchar(40) NOT NULL,
    Citta varchar(20) NOT NULL,
    Provincia char(2) NOT NULL,
    Via varchar(30) NOT NULL,
    NumeroCivico numeric NOT NULL,
    CAP numeric(5) NOT NULL,
    IndirizzoMail varchar(254) NOT NULL,
    NumeroTelefonico numeric(10) NOT NULL,
    TipoScuola char(2) NOT NULL,
    ComposizioneCircolo char(10) NOT NULL,
    FOREIGN KEY (ComposizioneCircolo)
    REFERENCES CIRCOLO (CodiceMeccanograficoCircolo)
    ON UPDATE CASCADE
    ON DELETE NO ACTION
)

CREATE TABLE DOCENTE_SCUOLA_MATERNA (
    CodiceInsegnanteMaterna numeric(4) PRIMARY KEY,
    CF char(16) NOT NULL,
```



```

Nome varchar(20) NOT NULL,
Cognome varchar(20) NOT NULL,
Sesso sesso,
DataDiNascita date NOT NULL
)

CREATE TABLE SUPPLENTE_MATERNA (
    CodiceSupplenteMaterna numeric(4) PRIMARY KEY,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL,
)

CREATE TABLE MAESTRA_DI_RUOLO_MATERNA (
    CodiceInsegnanteMaterna numeric(4) PRIMARY KEY,
    CF char(16) NOT NULL,
    IndirizzoMail varchar(254) NOT NULL,
    NumeroTelefonico numeric(10) NOT NULL,
    Stipendio smallint NOT NULL,
    FOREIGN KEY (CodiceInsegnanteMaterna)
    REFERENCES DOCENTE_SCUOLA_MATERNA (CodiceInsegnanteMaterna)
    ON UPDATE CASCADE
    ON DELETE SET NULL
)

CREATE TABLE DOCENTE_SCUOLA_ELEMENTARE (
    CodiceInsegnanteElementare numeric(4) PRIMARY KEY,
    CF char(16) NOT NULL,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL,
    Sesso sesso NOT NULL,
    DataDiNascita date NOT NULL
)

CREATE TABLE SUPPLENTE_ELEMENTARE (
    CodiceSupplenteElementare numeric(4) PRIMARY KEY,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL
)

CREATE TABLE MAESTRA_DI_RUOLO_ELEMENTARE (
    CodiceInsegnanteElementare numeric(4) PRIMARY KEY,
    IndirizzoMail varchar(254) NOT NULL,
    NumeroTelefonico numeric(10) NOT NULL,
    Stipendio numeric(9) NOT NULL,
    FOREIGN KEY (CodiceInsegnanteElementare)
    REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
)

```

```

        ON UPDATE CASCADE
        ON DELETE SET NULL
    )

CREATE TABLE SOSTEGNO (
    CodiceInsegnanteElementare numeric(4) PRIMARY KEY,
    IndirizzoMail varchar(254) NOT NULL,
    NumeroTelefonico char(20) NOT NULL,
    Stipendio numeric(9) NOT NULL,
    FOREIGN KEY (CodiceInsegnanteElementare)
    REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
    ON UPDATE CASCADE
    ON DELETE SET NULLL
)

CREATE TABLE ALUNNI (
    CodiceAlunno numeric(4) PRIMARY KEY,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    CF char(16) NOT NULL,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL,
    Sesso sesso NOT NULL,
    DataDiNascita date NOT NULL,
    Disabilita boolean NOT NULL,
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE SET NULL
,    UNIQUE (CodiceAlunno, CodiceMeccanograficoPlesso)
)

CREATE TABLE CLASSE (
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    Anno numeric(1) NOT NULL,
    Sezione char(1) NOT NULL,
    LezioneInPresenza boolean NOT NULL,
    PRIMARY KEY (Anno, Sezione, CodiceMeccanograficoPlesso),
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
CHECK (
    CodiceMeccanograficoPlesso = 'PGEE0100T' OR
    CodiceMeccanograficoPlesso = 'PGEE0101T' OR
    CodiceMeccanograficoPlesso = 'PGEE0102T' OR
    CodiceMeccanograficoPlesso = 'PGEE0103T' OR

```

```

        CodiceMeccanograficoPlesso = 'PGEE0104T' ),
CHECK (Anno = 1 OR
        Anno = 2 OR
        Anno = 3 OR
        Anno = 4 OR
        Anno = 5 )
)

CREATE TABLE MATERIA (
        CodiceMateria varchar(6) PRIMARY KEY,
        Nome varchar(15) NOT NULL UNIQUE
)

CREATE TABLE AFFIDATARIO (
        CodiceAffidatario char(16) PRIMARY KEY,
        Nome varchar(20) NOT NULL,
        Cognome varchar(20) NOT NULL,
        Sesso sesso NOT NULL,
        DataDiNascita date NOT NULL,
        TipoAffidatario char(10) NOT NULL,
        NumeroTelefonico char(20) NOT NULL
)

CREATE TABLE BAMBINI (
        CodiceBambino numeric(4) PRIMARY KEY,
        CodiceMeccanograficoPlesso char(10) NOT NULL,
        CF char(16) NOT NULL,
        Nome varchar(20) NOT NULL,
        Cognome varchar(20) NOT NULL,
        Sesso sesso NOT NULL,
        DataDiNascita date NOT NULL,
        FOREIGN KEY (CodiceMeccanograficoPlesso)
        REFERENCES PLESSO (CodiceMeccanograficoPlesso)
        ON UPDATE CASCADE
        ON DELETE NO ACTION,
CHECK (
        CodiceMeccanograficoPlesso = 'PGEE0105T' OR
        CodiceMeccanograficoPlesso = 'PGEE0106T' OR
        CodiceMeccanograficoPlesso = 'PGEE0107T' OR
        CodiceMeccanograficoPlesso = 'PGEE0108T' OR
        CodiceMeccanograficoPlesso = 'PGEE0109T'
)
UNIQUE ( CodiceBambino, CodiceMeccanograficoPlesso )
)

CREATE TABLE DATI_SANITARI_BAMBINO (
        Codice numeric(4) NOT NULL,

```

```

    Scuola char(10) NOT NULL,
    CF char(16) NOT NULL,
    Vaccini varchar(120) DEFAULT NULL,
    Allergie varchar(120) DEFAULT NULL,
    PRIMARY KEY (Codice, Scuola),
    FOREIGN KEY (Codice, Scuola)
    REFERENCES bambini(CodiceBambino, Scuola)
    ON UPDATE CASCADE
    ON DELETE CASCADE
)

CREATE TABLE DATI_ANAGRAFICI_ATA (
    CF char(16) PRIMARY KEY,
    Nome varchar(20) NOT NULL,
    Cognome varchar(20) NOT NULL,
    Sesso sesso NOT NULL,
    DataDiNascita date NOT NULL
)

CREATE TABLE DATI_LAVORATIVI_ATA (
    DatiAnagraficiATA char(16) PRIMARY KEY,
    IndirizzoMail varchar(254) NOT NULL,
    NumeroTelefonico char(20) NOT NULL,
    Stipendio smallint NOT NULL,
    TipoPersonale varchar(4) NOT NULL,
    Plesso char(10) NOT NULL,
    FOREIGN KEY (DatiAnagraficiATA)
    REFERENCES DATI_ANAGRAFICI_ATA (CF),
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (Plesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE NO ACTION
)

CREATE TABLE AFFIDAMENTO_BAMBINO (
    CodiceAffidatario char(16) NOT NULL,
    CodiceBambino numeric(4) NOT NULL,
    Scuola char(10) NOT NULL,
    FOREIGN KEY (CodiceAffidatario)
    REFERENCES AFFIDATARIO (CodiceAffidatario)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceBambino, Scuola)
    REFERENCES BAMBINI (CodiceBambino, CodiceMeccanograficoPlesso)
)

```

```

ON UPDATE CASCADE
ON DELETE CASCADE
)

```

```

CREATE TABLE AFFIDAMENTO_ALUNNO (
    CodiceAffidatario char(16) NOT NULL,
    CodiceAlunno numeric(4) default NULL,
    Scuola char(10) NOT NULL,
    FOREIGN KEY (CodiceAffidatario)
    REFERENCES AFFIDATARIO (CodiceAffidatario)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceAlunno, Scuola)
    REFERENCES ALUNNI (CodiceAlunno, CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE CASCADE
)

```

```

CREATE TABLE APPARTENENZA (
    CodiceAlunno numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    Anno numeric(1) NOT NULL,
    Sezione char(2) NOT NULL,
    FOREIGN KEY (CodiceAlunno, CodiceMeccanograficoPlesso)
    REFERENCES ALUNNI (CodiceAlunno, CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceMeccanograficoPlesso, Anno, Sezione)
    REFERENCES CLASSE (CodiceMeccanograficoPlesso, Anno,
    Sezione)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
    UNIQUE (CodiceAlunno, CodiceMeccanograficoPlesso)
)

```

```

CREATE TABLE ASSOCIAZIONE (
    CodiceInsegnanteElementare numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    Anno numeric(1) NOT NULL,
    Sezione char(1) NOT NULL,
    FOREIGN KEY (CodiceInsegnanteElementare)
    REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
)

```

```

        ON UPDATE CASCADE ON DELETE NO ACTION
    )

CREATE TABLE INSEGNAMENTO_PASSATO_ELEMENTARE (
    CodiceInsegnanteElementare numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    DataInizio date NOT NULL,
    DataFine date NOT NULL,
    FOREIGN KEY (CodiceInsegnanteElementare)
    REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare),
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
    PRIMARY KEY (CodiceInsegnanteElementare,
    CodiceMeccanograficoPlesso,
    DataInizio, DataFine)
)

CREATE TABLE INSEGNAMENTO_PRESENTE_ELEMENTARE (
    CodiceInsegnanteElementare numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    DataInizio date NOT NULL,
    FOREIGN KEY (CodiceInsegnanteElementare)
    REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
    PRIMARY KEY (CodiceInsegnanteElementare,
    CodiceMeccanograficoPlesso,
    DataInizio)
)

CREATE TABLE INSEGNAMENTO_PASSATO_MATERNA (
    CodiceInsegnanteMaterna numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    DataInizio date NOT NULL,
    DataFine date NOT NULL,
    FOREIGN KEY (CodiceInsegnanteMaterna)
    REFERENCES DOCENTE_SCUOLA_MATERNA (CodiceInsegnanteMaterna)
    ON UPDATE CASCADE

```

```

ON DELETE CASCADE,
FOREIGN KEY (CodiceMeccanograficoPlesso)
REFERENCES PLESSO (CodiceMeccanograficoPlesso)
ON UPDATE CASCADE
ON DELETE NO ACTION,
CHECK (DataFine > DataInizio AND
codicemeccanograficoplesso = 'PGEE0105T' OR
codicemeccanograficoplesso = 'PGEE0106T' OR
codicemeccanograficoplesso = 'PGEE0107T' OR
codicemeccanograficoplesso = 'PGEE0108T' OR
codicemeccanograficoplesso = 'PGEE0109T' )
PRIMARY KEY (CodiceInsegnanteMaterna,
CodiceMeccanograficoPlesso,
DataInizio)
)

CREATE TABLE INSEGNAMENTO_PRESENTE_MATERNA (
CodiceInsegnanteMaterna numeric(4) NOT NULL,
CodiceMeccanograficoPlesso char(10) NOT NULL,
DataInizio date NOT NULL,
FOREIGN KEY (CodiceInsegnanteMaterna)
REFERENCES DOCENTE_SCUOLA_MATERNA (CodiceInsegnanteMaterna)
ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY (CodiceMeccanograficoPlesso)
REFERENCES PLESSO (CodiceMeccanograficoPlesso)
ON UPDATE CASCADE
ON DELETE NO ACTION,
CHECK (
codicemeccanograficoplesso = 'PGEE0105T' OR
codicemeccanograficoplesso = 'PGEE0106T' OR
codicemeccanograficoplesso = 'PGEE0107T' OR
codicemeccanograficoplesso = 'PGEE0108T' OR
codicemeccanograficoplesso = 'PGEE0109T' )
PRIMARY KEY (CodiceInsegnanteMaterna,
CodiceMeccanograficoPlesso,
DataInizio)
)

CREATE TABLE SUPPLENZA_ELEMENTARE (
CodiceSupplenteElementare numeric(4) NOT NULL,
CodiceInsegnanteElementare numeric(4) NOT NULL,
DataInizio date NOT NULL,
DataFine date DEFAULT NULL,
FOREIGN KEY (CodiceSupplenteElementare)
REFERENCES SUPLENTE_ELEMENTARE (CodiceSupplenteElementare)

```

```

ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY (CodiceInsegnanteElementare)
REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
ON UPDATE CASCADE
ON DELETE CASCADE,
PRIMARY KEY (CodiceSupplenteElementare,
CodiceInsegnanteElementare, DataInizio)
)

```

```

CREATE TABLE SUPPLENZA_MATERNA (
CodiceSupplenteMaterna numeric(4) NOT NULL,
CodiceInsegnanteMaterna numeric(4) NOT NULL,
DataInizio date NOT NULL,
DataFine date NOT NULL,
FOREIGN KEY (CodiceSupplenteMaterna)
REFERENCES SUPLENTE_MATERNA (CodiceSupplenteMaterna)
ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY (CodiceInsegnanteMaterna)
REFERENCES DOCENTE_SCUOLA_MATERNA (CodiceInsegnanteMaterna)
ON UPDATE CASCADE
ON DELETE CASCADE,
PRIMARY KEY (CodiceSupplenteMaterna,
CodiceInsegnanteMaterna, DataInizio)
)

```

```

CREATE TABLE INSEGNAMENTO_MATERIA (
CodiceInsegnanteElementare numeric(4) NOT NULL,
Materia varchar(15) NOT NULL,
FOREIGN KEY (CodiceInsegnanteElementare)
REFERENCES DOCENTE_SCUOLA_ELEMENTARE (CodiceInsegnanteElementare)
ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY (Materia)
REFERENCES MATERIA (Nome)
ON UPDATE CASCADE
ON DELETE CASCADE,
UNIQUE (CodiceInsegnanteElementare, Materia)
)

```

```

CREATE TABLE INSEGNAMENTO_SOSTEGNO (
Sostegno numeric(4) NOT NULL,
CodiceAlunno numeric(4) NOT NULL,
CodiceMeccanograficoPlesso char(10) NOT NULL,
FOREIGN KEY (Sostegno)

```



```

REFERENCES SOSTEGNO (CodiceInsegnanteElementare)
ON UPDATE CASCADE
ON DELETE CASCADE,
FOREIGN KEY (CodiceAlunno,CodiceMeccanograficoPlesso)
REFERENCES ALUNNI (CodiceAlunno,CodiceMeccanograficoPlesso)
ON UPDATE CASCADE
ON DELETE CASCADE,
PRIMARY KEY (Sostegno, CodiceAlunno,
CodiceMeccanograficoPlesso)
)

CREATE TABLE VALUTAZIONE (
    Alunno numeric(4) NOT NULL,
    CodiceMeccanograficoPlesso char(10) NOT NULL,
    Materia varchar(6) NOT NULL,
    Voto varchar(15) NOT NULL,
    Semestre numeric(1) NOT NULL,
    FOREIGN KEY (Materia)
    REFERENCES MATERIA (CodiceMateria)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
    FOREIGN KEY (CodiceMeccanograficoPlesso)
    REFERENCES PLESSO (CodiceMeccanograficoPlesso)
    ON UPDATE CASCADE
    ON DELETE NO ACTION,
    PRIMARY KEY (CodiceAlunno, CodiceMeccanograficoPlesso,
Materia,Semestre)
)

```

### 3.2 Creazione dei vincoli non esprimibili

- *La valutazione viene fatta in decimi*

**ALTER TABLE VALUTAZIONE**

**ADD CHECK** ( VALUTAZIONE.Voto = '0' OR  
VALUTAZIONE.Voto = '1' OR VALUTAZIONE.Voto = '2' OR  
VALUTAZIONE.Voto = '3' OR VALUTAZIONE.Voto = '4' OR  
VALUTAZIONE.Voto = '5' OR VALUTAZIONE.Voto = '6' OR  
VALUTAZIONE.Voto = '7' OR VALUTAZIONE.Voto = '8' OR  
VALUTAZIONE.Voto = '9' OR VALUTAZIONE.Voto = '10' )

- *I semestri considerati in un anno scolastico sono due.*

**ALTER TABLE VALUTAZIONE**

**ADD CHECK** ( VALUTAZIONE.Semestre = 1 OR  
VALUTAZIONE.Semestre = 2 )

- *Un insegnante non può esercitare la propria professione contemporanea-  
mente in due scuole diverse nello stesso periodo.*

Con la ristrutturazione dello schema E-R , avvenuta in fase di progettazione logica, non c'è la necessità di implementare questo vincolo, a causa della separazione tra l'insegnamento presente e passato.

### 3.3 Creazione delle query

**OP 1 :** Iscrivere un nuovo alunno. (250 volte ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION aggiungi_alunno (  
  Alunno integer,  
  Plesso char(10),  
  CF_Alunno char(16),  
  Nome_Alunno varchar,  
  Cognome_Alunno varchar,  
  Sesso_Alunno char,  
  DataDiNascita_Alunno date,  
  Disabilita_Alunno boolean  
)  
RETURNS VOID  
LANGUAGE 'plpgsql'  
AS  
$$  
BEGIN  
  INSERT INTO ALUNNI (  
    CodiceAlunno,  
    CodiceMeccanograficoPlesso,  
    CF,  
    Nome,  
    Cognome,  
    Sesso,  
    DataDiNascita,  
    Disabilita  
  )  
  VALUES (  
    Alunno,  
    Plesso,  
    CF_Alunno,  
    Nome_Alunno,  
    Cognome_Alunno,  
    Sesso_Alunno,  
    DataDiNascita_Alunno,  
    Disabilita_Alunno  
  );  
END;  
$;
```

**OP 2 :** Aggiungere un nuovo bambino. (100 volte ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION Aggiungi_Bambino (  

```

```

codice_bambino numeric,
codice_scuola char(10),
CF char(16),
nome_bambino varchar,
cognome_bambino varchar,
sesso sesso,
data_di_nascita date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO BAMBINI (
CodiceBambino,
Scuola,
CF,
Nome,
Cognome,
Sesso,
DataDiNascita
)
VALUES (
codice_bambino,
codice_scuola,
CF,
nome_bambino,
cognome_bambino,
sesso,
data_di_nascita
);
END;
$$

```

**OP 3 :** Aggiungere un docente di scuola primaria. (1 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION Aggiungi_Docente_Scuola_Elementare(
CodiceInsegnanteElementare numeric,
CF char,
Nome varchar(20),
Cognome varchar(20),
Sesso sesso,
DataDiNascita date,
Data_Inizio date,
Plesso char(10),

```

```

Mail varchar,
Telefono numeric(10),
Stipendio numeric
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO DOCENTE_SCUOLA_ELEMENTARE
VALUES (
CodiceInsegnanteElementare,
CF,
Nome,
Cognome,
Sesso,
DataDiNascita
) ;

INSERT INTO Maestra_di_ruolo_elementare (
codiceinsegnanteelementare,
indirizzomail,
numerotelefonico,
stipendio
)
VALUES
(CodiceInsegnanteElementare,
Mail,
Telefono,
Stipendio
);
INSERT INTO Insegnamento_presente_elementare (
CodiceInsegnanteElementare,
Codicemeccanograficoplesso,
Datainizio
)
VALUES
(
CodiceInsegnanteElementare,
Plesso,
Data_Inizio
);
END;
$$;

```

**OP 4 :** Aggiungere un affidatario. (2 volte all'inizio del ciclo scolastico)

**Elementare:**

```
CREATE OR REPLACE FUNCTION
aggiungi_affidatario_elementare (
codice_aff character(16),
nome_aff varchar,
cognome_aff varchar,
sesso sesso,
data_di_nascita date,
tipo_aff char,
numero_telefono_aff numeric,
alunno numeric(4),
scuola char(10)
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO AFFIDATARIO (
CodiceAffidatario,
Nome,
Cognome,
Sesso,
DataDiNascita,
TipoAffidatario,
NumeroTelefonico
)
VALUES (
codice_aff,
nome_aff,
cognome_aff,
sesso,
data_di_nascita,
tipo_aff,
numero_telefono_aff
);
INSERT INTO affidamento_alunno (
CodiceAffidatario,
CodiceAlunno,
Scuola
)
VALUES
( codice_aff,
alunno,
```

```

scuola
);
END;
$$

```

**Materna:**

```

CREATE OR REPLACE FUNCTION
aggiungi_affidatario_materna (
codice_aff character(16),
nome_aff varchar,
cognome_aff varchar,
 Sesso Sesso,
data_di_nascita date,
tipo_aff char,
numero_telefono_aff numeric,
bambino numeric(4),
scuola char(10)
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO AFFIDATARIO (
CodiceAffidatario,
Nome,
Cognome,
Sesso,
DataDiNascita,
TipoAffidatario,
NumeroTelefonico
)
VALUES (
codice_aff,
nome_aff, cognome_aff, Sesso,
data_di_nascita,
tipo_aff,
numero_telefono_aff
);
INSERT INTO affidamento_bambino (
CodiceAffidatario,
CodiceBambino,
Scuola
)
VALUES
(

```

```

codice_aff,
bambino,
scuola
);
END;
$$

```

**OP 5 :** Aggiungere un dipendente del personale ATA. (1 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
aggiunta_personale_ATA (
CF_ATA char(16),
Nome_ATA varchar,
Cognome_ATA varchar,
Sesso_ATA sesso,
DataDiNascita_ATA date,
IndirizzoMail_ATA varchar,
NumeroTelefonico_ATA char,
Stipendio_ATA smallint,
TipoPersonale_ATA varchar,
Plesso_ATA char
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO dati_anagrafici_ATA (
CF,
Nome,
Cognome,
Sesso,
DataDiNascita
)
VALUES (
CF_ATA,
Nome_ATA,
Cognome_ATA,
Sesso_ATA,
DataDiNascita_ATA
);
INSERT INTO dati_lavorativi_ATA (
IndirizzoMail,
NumeroTelefonico,
Stipendio,

```



```

TipoPersonale,
DatiAnagraficiATA,
Plesso
)
VALUES (
IndirizzoMail_ATA,
NumeroTelefonico_ATA,
Stipendio_ATA,
TipoPersonale_ATA,
CF_ATA,
Plesso_ATA
);
END;
$$;

```

**OP 6 :** Assegnare una nuova supplente di scuola elementare. (1 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION Aggiungi_Supplente_Elementare (
codice_supplente_elementare numeric,
nome_supplente varchar,
cognome_supplente varchar
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO supplente_elementare (
CodiceSupplenteElementare,
Nome,
Cognome
)
VALUES (
codice_supplente_elementare,
nome_supplente,
cognome_supplente
);
END;
$$

```

**OP 7 :** Aggiungere una maestra di ruolo di scuola materna. (1 volta all'anno)

```

CREATE OR REPLACE FUNCTION Aggiungi_maestra_materna (
codice_insegnante_materna numeric,
cf_materna char(10),

```

```

nome_materna varchar,
cognome_materna varchar,
sesso_materna sesso,
data_d_nascita_materna date,
plesso char(10)
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO docente_scuola_materna
VALUES (
codice_insegnante_materna,
cf_materna,
nome_materna,
cognome_materna,
sesso_materna,
data_di_nascita_materna
);
END;
$$

```

**OP 8 :** Assegnare una valutazione a un alunno. (2 volte ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
aggiungi_valutazione_alunno (
codiceAlunno numeric,
codicePlesso char,
materia varchar,
valutazione varchar,
semestre numeric
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO VALUTAZIONE(
Alunno,
CodiceMeccanograficoPlesso,
Materia,
Voto,
Semestre
)
VALUES (

```

```

codiceAlunno,
codicePlesso,
materia,
valutazione,
semestre
);
END;
$$

```

**OP 9 :** Eliminare gli alunni che hanno concluso il ciclo scolastico. (250 volte ogni anno scolastico)

Per evitare il problema di eliminare gli alunni della quarta che si iscriveranno in quinta elementare, si devono prima eliminare gli ex-alunni della quinta elementare.

```

CREATE OR REPLACE FUNCTION Delete_alunno ()
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$ BEGIN
DELETE FROM VALUTAZIONE
WHERE VALUTAZIONE.Alunno IN (SELECT CodiceAlunno
FROM Appartenenza
WHERE Anno = 5);
DELETE FROM ALUNNI
WHERE ALUNNI.CodiceAlunno IN (SELECT CodiceAlunno
FROM Appartenenza
WHERE Anno = 5);
END;
$$

```

**OP 10 :** Eliminare una maestra. (1 volta ogni cinque anni)

**Elementare :**

```

CREATE OR REPLACE FUNCTION Delete_maestra_elementare (
codice_maestra numeric,
data_fine date,
codice_plesso char,
data_inizio date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN

```

```

DELETE FROM maestra_di_ruolo_elementare
WHERE maestra_di_ruolo_elementare.CodiceInsegnanteElementare
= codice_maestra;
DELETE FROM insegnamento_presente_elementare
WHERE insegnamento_presente_elementare.CodiceInsegnanteElementare
= codice_maestra;
INSERT INTO insegnamento_passato_elementare (
codiceinsegnanteelementare,
codicemeccanograficoplesso,
datainizio,
datafine
)
VALUES
(codice_maestra,
codice_plesso,
data_inizio,
data_fine);
END;
$$

```

#### **Materna :**

```

CREATE OR REPLACE FUNCTION
delete_maestra_materna (
codice_maestra numeric,
codice_plesso char,
data_inizio date,
data_fine date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
DELETE FROM maestra_di_ruolo_materna
WHERE maestra_di_ruolo_materna.CodiceInsegnanteMaterna
= codice_maestra;
DELETE FROM insegnamento_presente_materna
WHERE insegnamento_presente_materna.CodiceInsegnanteMaterna
= codice_maestra;
INSERT INTO insegnamento_passato_materna (
CodiceInsegnanteMaterna,
CodiceMeccanograficoPlesso,
DataInizio,
DataFine
)
VALUES

```

```

(codice_maestra,
codice_plesso,
data_inizio,
data_fine
);
END;
$$

```

**OP 11** : Sostituire la preside. (1 volta ogni cinque anni)

```

CREATE OR REPLACE FUNCTION
replace_preside (
cf_ex_Preside char,
cf_preside char,
nome_preside varchar,
cognome_preside varchar,
 Sesso_preside sesso,
data_di_nascita_preside date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO dirigente_scolastico (
CF,
Nome,
Cognome,
Sesso,
DataDiNascita
)
VALUES
(
cf_preside,
nome_preside,
cognome_preside,
 Sesso_preside,
data_di_nascita_preside
);
UPDATE CIRCOLO
SET DirigenteScolastico = cf_preside
WHERE CIRCOLO.CodiceMeccanograficoCircolo = 'PGEE0510T';
DELETE FROM dirigente_scolastico
WHERE dirigente_Scolastico.cf = cf_ex_preside;
END;
$$

```

**OP 12 :** Sostituire una docente di ruolo con una supplente. (2 volte a semestre)

**Elementare :**

```
CREATE OR REPLACE FUNCTION
substitution_elementare (
codice_maestra numeric,
codice_supplente numeric,
data_inizio date,
data_fine date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO supplenza_elementare (
CodiceSupplenteElementare,
CodiceInsegnanteElementare,
DataInizio,
DataFine
)
VALUES
(
codice_supplente,
codice_maestra,
data_inizio,
data_fine
);
END;
$$
```

**Materna :**

```
CREATE OR REPLACE FUNCTION
substitution_materna (
codice_maestra numeric,
codice_supplente numeric,
data_inizio date,
data_fine date
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
INSERT INTO supplenza_materna (
```

```

CodiceSupplenteMaterna,
CodiceInsegnanteMaterna,
DataInizio,
DataFine
)
VALUES
(
codice_supplente,
codice_maestra,
data_inizio,
data_fine
);
END;
$$

```

**OP 13 :** Inserire le allergie di un bambino già iscritto alla scuola. (1 volta ogni ciclo scolastico)

```

CREATE OR REPLACE FUNCTION
insert_allergy (
codice_bambino numeric,
codice_scuola char,
allergie_bambino varchar
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
UPDATE dati_sanitari_bambino
SET Allergie = allergie_bambino
WHERE dati_sanitari_bambino.Codice = codice_bambino AND
dati_sanitari_bambino.Scuola = codice_scuola;
END;
$$

```

**OP 14 :** Assegnare una lezione in presenza o online a una classe. (1 volta ogni settimana)

**Lezione in Presenza :**

```

CREATE OR REPLACE FUNCTION
lezione_in-presenza (
codice_scuola char,
anno_scuola numeric,
sezione_scuola char
)

```

```

RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
UPDATE CLASSE
SET LezioneInPresenza = true
WHERE Classe.CodiceMeccanograficoPlesso = codice_scuola AND
Classe.Anno = anno_scuola AND Classe.Sezione = sezione_scuola;
END;
$$

```

**Lezione Online :**

```

CREATE OR REPLACE FUNCTION
lezione_online (
codice_scuola char,
anno_scuola numeric,
sezione_scuola char
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
UPDATE CLASSE
SET LezioneInPresenza = false
WHERE CLASSE.CodiceMeccanograficoPlesso = codice_scuola AND
Classe.Anno = anno_scuola AND Classe.Sezione = sezione_scuola;
END;
$$

```

**OP 15 :** Assegnare una materia al docente. (2 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
insegnante_materia (
codice_insegnante numeric,
materia_insert varchar(6)
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
UPDATE insegnamento_materia
SET materia = materia_insert
WHERE insegnamento_materia.CodiceInsegnanteElementare =

```



```

codice_insegnante;
END;
$$

```

**OP 16 :** Aggiornare lo stipendio del personale ATA. (1 volta ogni 5 anni)

```

CREATE OR REPLACE FUNCTION
aggiornamento_stipendio_ata (
tipo_personale varchar,
stipendio smallint
)
RETURNS VOID
LANGUAGE 'plpgsql'
AS
$$
BEGIN
UPDATE dati_lavorativi_ata
SET dati_lavorativi_ata.Stipendio = stipendio
WHERE dati_lavorativi_ata.TipoPersonale = tipo_personale;
END;
$$

```

**OP 17 :** Stampare i voti della pagella di un alunno. (2 volte ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
stampa_voti_pagella (
alunnopagella numeric,
plesso char,
semestrepagella numeric
)
RETURNS TABLE (materia varchar, voto varchar)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT M.Nome, V.Voto
FROM VALUTAZIONE V JOIN MATERIA M ON V.materia = M.codicemateria
WHERE V.Alunno = alunnopagella AND
V.CodiceMeccanograficoPlesso = plesso AND
V.Semestre = semestrepagella
);
END;
$BODY$;

```

**OP 18 :** Stampare i docenti di una classe. (1 volta ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION
stampa_docenti_di_una_classe (
codice_meccanografico_plesso char,
annoo numeric,
sezione char)
RETURNS TABLE (
CodiceInsegnanteElementare numeric,
Nome varchar,
Cognome varchar
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT ASSOCIAZIONE.CodiceInsegnanteElementare, docente_scuola_elementare.nome,
docente_scuola_elementare.cognome
FROM ASSOCIAZIONE JOIN docente_scuola_elementare ON
ASSOCIAZIONE.CodiceInsegnanteElementare =
docente_scuola_elementare.CodiceInsegnanteElementare
WHERE codice_meccanografico_plesso =
ASSOCIAZIONE.CodiceMeccanograficoPlesso AND
annoo = ASSOCIAZIONE.Anno AND
sezione = ASSOCIAZIONE.Sezione
);
END;
$BODY$;
```

**OP 19 :** Stampare gli alunni di una classe. (1 volta ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION
Stampa_alunni_di_una_classe (
plesso char,
annoplesso numeric,
sezioneplesso char
)
RETURNS TABLE (
Nome varchar,
Cognome varchar
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
```

```

RETURN QUERY (
SELECT ALUNNI.Nome, ALUNNI.Cognome
FROM APPARTENENZA, ALUNNI
WHERE plesso = APPARTENENZA.codiceMeccanograficoPlesso AND
plesso = ALUNNI.codiceMeccanograficoPlesso AND
APPARTENENZA.CodiceAlunno = ALUNNI.CodiceAlunno AND
annoplesso = APPARTENENZA.Anno AND
sezioneplesso = APPARTENENZA.Sezione
);
END;
$BODY$;

```

**OP 20** : Stampare i dati lavorativi del personale ATA del circolo. (1 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
stampa_dati_lavorativi_personale_ata ()
RETURNS TABLE (
cf_ata char,
IndirizzoMail varchar,
NumeroTelefonico char,
Stipendio smallint,
TipoPersonale varchar,
Plesso char
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DLA.DatiAnagraficiATA, DLA.IndirizzoMail,
DLA.NumeroTelefonico, DLA.Stipendio,
DLA.TipoPersonale, DLA.Plesso
FROM DATI_LAVORATIVI_ATA DLA, DATI_ANAGRAFICI_ATA DAA, PLESSO
P
WHERE DLA.DatiAnagraficiATA = DAA.CF AND
DLA.Plesso = P.CodiceMeccanograficoPlesso
);
END;
$BODY$;

```

**OP 21** : Stampare nome, cognome e codice insegnante di tutti i docenti di un plesso. (1 volta ogni anno per ogni circolo)

**Elementare :**

```

CREATE OR REPLACE FUNCTION

```

```

stampa_docenti_elementare_plesso (
codice_meccanografico_plesso char
)
RETURNS TABLE (
Nome varchar,
Cognome varchar,
CodiceInsegnanteElementare numeric
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DOCENTE_SCUOLA_ELEMENTARE.Nome,
DOCENTE_SCUOLA_ELEMENTARE.Cognome,
DOCENTE_SCUOLA_ELEMENTARE.CodiceInsegnanteElementare
FROM DOCENTE_SCUOLA_ELEMENTARE JOIN
INSEGNAMENTO_PRESENTE_ELEMENTARE ON
DOCENTE_SCUOLA_ELEMENTARE.CodiceInsegnanteElementare =
INSEGNAMENTO_PRESENTE_ELEMENTARE.CodiceInsegnanteElementare
WHERE CodiceMeccanograficoPlesso = codice_meccanografico_plesso
);
END;
$BODY$;

```

#### **Materna :**

```

CREATE OR REPLACE FUNCTION
stampa_docenti_materna_plesso (
codice_meccanografico_plesso char
)
RETURNS TABLE (
Nome varchar,
Cognome varchar,
CodiceInsegnanteMaterna numeric
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DOCENTE_SCUOLA_MATERNA.Nome,
DOCENTE_SCUOLA_MATERNA.Cognome,
DOCENTE_SCUOLA_MATERNA.CodiceInsegnanteMaterna
FROM DOCENTE_SCUOLA_MATERNA JOIN INSEGNAMENTO_PRESENTE_MATERNA
ON
DOCENTE_SCUOLA_MATERNA.CodiceInsegnanteMaterna =

```

```

INSEGNAMENTO_PRESENTE_MATERNA.CodiceInsegnanteMaterna
WHERE CodiceMeccanograficoPlesso = codice_meccanografico_plesso
);
END;
$BODY$

```

**OP 22** : Stampare i codici insegnante di tutti i docenti del circolo. (1 volta ogni anno per ogni circolo)

```

CREATE OR REPLACE FUNCTION
Stampa_codici_insegnanti_circolo ()
RETURNS TABLE (
CodiceInsegnanteELEMENTARE numeric,
CodiceInsegnanteMATERNA numeric
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DE.CodiceInsegnanteElementare, DM.CodiceInsegnanteMaterna
FROM DOCENTE_SCUOLA_ELEMENTARE AS DE,
DOCENTE_SCUOLA_MATERNA AS DM,
INSEGNAMENTO_PRESENTE_ELEMENTARE AS IPE,
INSEGNAMENTO_PRESENTE_MATERNA AS IPM
WHERE DE.CodiceInsegnanteElementare =
IPE.CodiceInsegnanteElementare AND
DM.CodiceInsegnanteMaterna = IPM.CodiceInsegnanteMaterna
AND IPE.CodiceMeccanograficoPlesso =
ANY (SELECT PLESSO.CodiceMeccanograficoPlesso
FROM PLESSO)
);
END;
$BODY$;

```

**OP 23** : Stampare nome, cognome, scuola e classe di uno studente. (1250 volte ogni semestre)

```

CREATE OR REPLACE FUNCTION
stampa_dati_alunno (
codice_meccanografico_plesso char,
codice_alunno numeric
)
RETURNS TABLE (
Nome varchar,
Cognome varchar,

```

```

CodiceMeccanograficoPlesso char,
Anno numeric,
Sezione char
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT ALUNNI.Nome, ALUNNI.Cognome,
APPARTENENZA.CodiceMeccanograficoPlesso, APPARTENENZA.Anno,
APPARTENENZA.Sezione
FROM ALUNNI, APPARTENENZA
WHERE ALUNNI.CodiceAlunno = APPARTENENZA.CodiceAlunno AND
ALUNNI.CodiceMeccanograficoPlesso =
APPARTENENZA.CodiceMeccanograficoPlesso AND
ALUNNI.CodiceAlunno = codice_alunno AND
APPARTENENZA.CodiceMeccanograficoPlesso =
codice_meccanografico_plesso
);
END;
$BODY$;

```

**OP 24** : Stampare gli studenti che non frequentano IRC. (1 volta ogni anno scolastico)

```

CREATE OR REPLACE FUNCTION
stampa_non_frequentanti_religione ()
RETURNS TABLE (
Nome character varying(20),
Cognome character varying(20),
CodiceAlunno numeric(4))
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DISTINCT ALUNNI.Nome, ALUNNI.Cognome, ALUNNI.CodiceAlunno
FROM VALUTAZIONE JOIN ALUNNI ON
ALUNNI.CodiceAlunno = VALUTAZIONE.Alunno
WHERE VALUTAZIONE.Materia = 'REG08' AND
VALUTAZIONE.Voto = '0'
);
END;
$BODY$;

```

**OP 25** : Stampare i dati anagrafici del personale ATA. (1 volta ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION
stampa_dati_anagrafici_personale_ata ()
RETURNS TABLE (
CF char,
Nome varchar,
Cognome varchar,
Sesso sesso,
DataDiNascita date
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT *
FROM DATI_ANAGRAFICI_ATA
);
END;
$BODY$;
```

**OP 26** : Stampare tutte le supplenti che hanno sostituito le docenti presenti e/o passate. (1 volta ogni anno scolastico)

```
CREATE OR REPLACE FUNCTION stampa_supplenti ()
RETURNS TABLE (
Codice_supplente numeric,
Nome varchar,
Cognome varchar
)
LANGUAGE 'plpgsql'
AS
$BODY$
BEGIN
RETURN QUERY (
SELECT DISTINCT S.CodiceSupplenteElementare, S.Nome, S.Cognome
FROM SUPPLENTE_ELEMENTARE S JOIN SUPPLENZA_ELEMENTARE E ON
E.CodiceSupplenteElementare = S.CodiceSupplenteElementare
);
END;
$BODY$;
```

## 4 Esecuzione delle operazioni principali

**OP 1 :** Iscrivere un nuovo alunno. (250 volte ogni anno scolastico)

```
SELECT (Aggiungi_Alunno(1, 'PGEE0100T', 'SXTJFN57L44M308D',
'Michela', 'Tommasini', 'F', '06-09-2012', 'false'));
```

	codicealunno [PK] numeric (4)	codicemeccanograficoplesso character (10)	cf character (16)	nome character varying (20)	cognome character varying (20)	sex character (1)	datadinsc date	disabilita boolean
1	1	PGEE0100T	SXTJFN57L44M30...	Michela	Tommasini	F	2012-09-06	false
2	2	PGEE0100T	DJNXDL49M25L27...	Filippo	Blasi	M	2013-01-16	false
3	3	PGEE0100T	WMCYD093C22F1...	Francesco	Rutili	M	2014-03-05	false
4	4	PGEE0100T	CFBTW81B56L53...	Alessandra	Portici	F	2015-12-25	false
5	5	PGEE0100T	CFBTW81B56L53...	Sofia	Domizi	F	2016-10-20	false
6	6	PGEE0100T	TQUNPT58P03M05...	Leonardo	Battaglia	M	2012-11-17	false
7	7	PGEE0100T	YPFHB99P152902J	Martina	Ricci	F	2013-01-10	false
8	8	PGEE0100T	DXDYKX88L45E98F	Damiano	Riccardi	M	2014-02-08	false
9	9	PGEE0100T	ZVCMGY90B59D01...	Gaia	Santarelli	F	2015-12-28	true
10	10	PGEE0100T	GHFRL91M42I26...	Stefano	Belli	M	2016-10-20	true
11	11	PGEE0100T	YSSNFG74S08A08...	Veronica	Carletti	F	2012-11-15	true
12	12	PGEE0100T	NZLFST58H55C99...	Francesco	Barletta	M	2013-01-25	true

**OP 2 :** Aggiungere un nuovo bambino. (100 volte ogni anno scolastico)

```
SELECT (Aggiungi_Bambino(1, 'PGEE0109T', 'XSVFBV61R65D261G',
'Alessandra', 'Annibali', 'F', '06-10-2017'));
```

	codicebambino [PK] numeric (4)	codicemeccanograficoplesso character (10)	cf character (16)	nome character varying (20)	cognome character varying (20)	sex character (1)	datadinsc date
1	1	PGEE0109T	XSVFBV61R65D26...	Alessandra	Annibali	F	2017-10-06
2	2	PGEE0109T	TPQKKU58S28E86...	Valeria	Ticini	M	2018-11-02
3	3	PGEE0109T	SHMTNL83A07F03...	Francesca	Metelli	F	2019-10-22
4	4	PGEE0109T	TJHZVT40A58C676L	Lorenzo	Savini	M	2017-11-07
5	5	PGEE0109T	ZJMGSB69H46E29...	Naide	Cesarini	F	2018-01-27
6	6	PGEE0109T	QHSJQP70H58F42...	Vincenzo	Tempesta	M	2019-11-09
7	7	PGEE0109T	CHMTGD84C10G7...	Vittoria	Cruciani	F	2017-12-29
8	8	PGEE0109T	FZCKTM75T16A77...	Lorenzo	Proietti	M	2018-07-02
9	9	PGEE0109T	QQMHSL55C17F56...	Lorenza	Ortolani	F	2019-06-22
10	10	PGEE0109T	ZMDTTS61P63E55...	Dario	Rossi	M	2017-06-12
11	11	PGEE0109T	FFGMTT58R16E46...	Carla	Bianchi	F	2018-04-01
12	12	PGEE0109T	MDSRRN55H19H5...	Francesco	Supremi	M	2019-02-21
13	13	PGEE0109T	YVSDRL64P65H17...	Giulia	Premi	F	2017-01-11



**OP 3 :** Aggiungere un docente di scuola elementare. (1 volta ogni anno scolastico)

```
SELECT (Aggiungi_Docente_Scuola_Elementare(3031, 'GLLMTN70H62A202U',
'Martina', 'Galli', 'F', '22/06/1970', '09/09/2021', 'PGEE0100T',
'martinagalli@gmail.com', '3421567981', '1300'));
```

Tabella “docente\_scuola\_elementare”

	codiceinsegnanteelementare [PK] numeric (4)	cf character (16)	nome character varying (20)	cognome character varying (20)	sex character (1)	datadinascita date
1	3031	GLLMTN70H62A20...	Martina	Galli	F	1970-06-22
2	3032	SCLLNZ77C13A111I	Lorenzo	Siciliani	M	1977-03-13
3	3033	MRTLSTN74R47I921K	Alessandra	Martini	F	1974-07-06
4	3034	LZRNPF96E07H97...	Vittoria	Bianchini	F	1972-02-02
5	3035	ZMQSCC68L23L87...	Alessia	Todini	F	1973-07-06
6	3036	MFJQQV84T71L75...	Marcello	Sensini	M	1971-07-25
7	3037	NLDMCT32C07F77...	Chiara	Tedeschi	F	1978-12-02
8	3038	RCTNBL77T66D34...	Danilo	Sensi	M	1972-10-12
9	3039	PZPMDJ42S67E38...	Antonella	Marchi	F	1975-11-22
10	3040	JPBDPP93B64I555F	Francesco	Piantoni	M	1977-12-29
11	3041	ZMBKNU36H70F22...	Marcella	Pisani	F	1979-04-27
12	3042	FBHGRE91B52B31...	Mauro	Martinelli	M	1974-02-05
13	3043	FPAHCC79C23M10...	Claudia	Cantiani	F	1977-01-17

Tabella “insegnamento\_presente\_elementare”

	codiceinsegnanteelementare [PK] numeric (4)	codicemeccanograficoplesso [PK] character (10)	datainizio [PK] date
1	3031	PGEE0100T	2021-09-09
2	3032	PGEE0100T	2018-09-09
3	3033	PGEE0100T	2017-09-01
4	3034	PGEE0100T	2015-09-10
5	3035	PGEE0100T	2012-09-12
6	3036	PGEE0100T	2013-09-09
7	3037	PGEE0100T	2020-09-09
8	3038	PGEE0100T	2015-09-09
9	3039	PGEE0100T	2016-09-09
10	3040	PGEE0100T	2009-09-10
11	3041	PGEE0100T	2020-09-12
12	3042	PGEE0100T	2002-09-05
13	3043	PGEE0100T	2016-09-06

Tabella “maestra\_di\_ruolo\_elementare”

	codiceinsegnanteelementare [PK] numeric (4)	indirizzomail character varying (254)	numerotelefonico numeric (10)	stipendio numeric (9)
1	3031	martinagalli@gmail.com	3421567981	1300
2	3032	lorenzodiciliani@gmail.com	3429345115	1300
3	3033	alessandramartini@gmail.com	3775060904	1300
4	3034	vittoriabianchini@gmail.com	3423548798	1300
5	3035	alessiatodini@gmail.com	3423425287	1300
6	3036	marcellosensini@gmail.com	3423481651	1300
7	3037	chiaratedeschi@gmail.com	3421566987	1300
8	3038	danilosensi@gmail.com	3421616558	1300
9	3039	antonellamarchi@gmail.com	3416567981	1300
10	3040	francescopiantoni@gmail.com	3423552512	1300
11	3041	marcellapisani@gmail.com	3421321157	1300
12	3042	mauromartinelli@gmail.com	3423546526	1300
13	3043	claudiacantiani@gmail.com	3423262637	1300

**OP 8 :** Assegnare una valutazione ad un alunno. (2 volte ogni anno scolastico)

```
SELECT (aggiungi_valutazione_alunno ('1', 'PGEE0100T', 'REG01',
'9', '1'));
```

	alunno [PK] numeric (4)	codicemeccanograficoplesso [PK] character (10)	materia [PK] character varying (6)	voto character varying (15)	semestre [PK] numeric (1)
1	1	PGEE0100T	REG01	9	1
2	1	PGEE0100T	REG01	9	2
3	1	PGEE0100T	REG02	9	1
4	1	PGEE0100T	REG02	9	2
5	1	PGEE0100T	REG03	9	1
6	1	PGEE0100T	REG03	9	2
7	1	PGEE0100T	REG04	9	1
8	1	PGEE0100T	REG04	9	2
9	1	PGEE0100T	REG05	9	1
10	1	PGEE0100T	REG05	9	2
11	1	PGEE0100T	REG06	9	1
12	1	PGEE0100T	REG06	8	2
13	1	PGEE0100T	REG07	9	1

**OP 18 :** Stampare i voti della pagella di un alunno. (2 volte ogni anno scolastico)

```
SELECT (stampa_voti_pagella( 1, 'PGEE0100T', 1))
```

	stampa_voti_pagella record	
1	(Italiano,9)	
2	(Matematica,9)	
3	(Storia,9)	
4	(Geografia,9)	
5	(Inglese,9)	
6	(Scienze,9)	
7	(Arte,9)	
8	(Religione,9)	
9	(Ed.Fisica,9)	
10	(Musica,9)	

**OP 20 :** Stampare gli alunni di una classe. (1 volta ogni anno scolastico)

```
SELECT (Stampa_alunni_di_una_classe ('PGEE0100T', 1, 'A'))
```

	stampa_alunni_di_una_classe record	
1	(Riccardo,Marini)	
2	(Sofia,Domizi)	
3	(Stefano,Belli)	
4	(Giulia,Nicolini)	
5	(Giada,Gregori)	
6	(Carla,Savona)	
7	(Angelo,Poretti)	
8	(Gioele,Simonetti)	

**OP 21 :** Stampare gli addetti del personale ATA del circolo. (1 volta ogni anno scolastico)

```
SELECT stampa_dati_lavorativi_personale_ata ();
```

	stampa_dati_lavorativi_personale_ata	
	record	
1	(MCSDHP98B65C324F,galabianchi@gmail.com,"34714165",1200,puli,"PGEE0101T ")	
2	(ZVTZVK84C01D940E,gloriapollici@gmail.com,"34719797",1200,mens,"PGEE0102T ")	
3	(DOUKHT27C55C290F,galaclerici@gmail.com,"34712965",1200,appl,"PGEE0100T ")	
4	(DVGTMJ38D57H063Y,mariarodi@gmail.com,"34719892",1200,appl,"PGEE0100T ")	
5	(FBYCMS45T21D269L,marcosangiorgio@gmail.com,"34712153",1200,puli,"PGEE0100T ")	
6	(FLNRQX58R16M188S,leonardoproietti@gmail.com,"34712369",1200,puli,"PGEE0100T ")	
7	(GRHWNS98R41G359H,lorenzosaveri@gmail.com,"34712917",1200,mens,"PGEE0100T ")	
8	(GSPSNG53M44E473Y,valeriamartini@gmail.com,"34711563",1250,mens,"PGEE0100T ")	
9	(HLWBSM50P50H516D,saveriolori@gmail.com,"34713648",1200,coll,"PGEE0101T ")	
10	(HTHPMH30E02E844U,virginiapiermarini@gmail.com,"34711238",1200,coll,"PGEE0101T ")	
11	(HWCFCX38B65Z222R,emanuelesarri@gmail.com,"34711465",1200,segr,"PGEE0101T ")	
12	(LLYTCS78T23I082V,lorettapanella@gmail.com,"34716515",1200,segr,"PGEE0101T ")	
13	(LMBLGU73L28A301K,luigilombardi@gmail.com,"34795164",1200,appl,"PGEE0101T ")	

**OP 23 :** Stampare nome, cognome, scuola e classe di uno studente. (1.250 volte ogni semestre)

```
SELECT (stampa_dati_alunno ('PGEE0100T', 1));
```

	stampa_dati_alunno	
	record	
1	(Michela,Tommasini,"PGEE0100T ",5,"A ")	