UNIVERSITÀ DELLA CALABRIA

Facoltà di SS.MM.FF.NN.

Corso di Laurea in Informatica

A.A. 2019/2020

PROGETTO PER IL CORSO DI

BASI DI DATI RELAZIONALI

DOCENTE: PROF. P. RULLO

LABORATORIO: ING. G. LABOCCETTA,
DOTT.SSA D. ANGILICA

SISTEMA INFORMATIVO

PER LA GESTIONE DI

UN ISTITUTO COMPRENSIVO.

Gruppo 23
189714 Francesco Esposito
191857 Davide Gena
189676 Antonio Talarico

1. Tematica Progettuale

La progettazione del sistema informatico in esame riguarda la gestione di un istituto comprensivo.

2. Raccolta e Analisi dei Requisiti

2.1. Raccolta dei Requisiti

	Requisiti Richiesti
1	L'istituto comprensivo StudioOnline vuole riorganizzare il proprio database a seguito
2	di alcuni aggiornamenti nella normativa.
3	Il complesso scolastico include tre gradi: scuola dell'infanzia, scuola primaria e scuola
4	secondaria inferiore. Ciascuna scuola è distribuita in diversi plessi e ciascun plesso
5	ospita anche più di un grado scolastico.
6	Le iscrizioni avvengono nel mese di gennaio: i genitori iscrivono i propri figli e di
7	ognuno di loro viene registrata l'anagrafica.
8	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad uno dei 3 gradi
9	dell'istituto ma si vuole memorizzare per ciascun ragazzo lo storico delle iscrizioni.
10	All'atto dell'iscrizione, ogni studente è associato ad una classe.
11	I plessi sono identificati da un indirizzo e sono composti da delle aule. Di ogni aula si
12	conoscono i metri quadri, hanno un numero massimo di studenti ospitabili e la fascia
13	di età degli studenti ospitabili.
14	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero (1-4 per
15	l'infanzia, 1-5 per la primaria e 1-3 per le medie) e da una lettera dell'alfabeto (aule
16	di gradi diversi possono avere la stessa coppia). Di ciascun insegnante si conosce,
17	ogni anno, il numero di ore lavorative settimanali e le classi a cui è assegnato.

2.2. Analisi dei Requisiti.

2.2.1. Eliminazione delle Ambiguità.

Linea	Termine	Sinonimi	Motivazione Correzione
3	complesso scolastico	istituto comprensivo	sinonimo
8	studente	ragazzo	sinonimo
4	Scuola secondaria inferiore	medie	sinonimo
4	Scuola	grado	sinonimo

2.2.2. RISTRUTTURAZIONE DEI REQUISITI RICHIESTI

	SPECIFICHE RISTRUTTURATE
1	Il complesso scolastico StudioOnline vuole riorganizzare il proprio database a seguito
2	di alcuni aggiornamenti nella normativa.
3	Il complesso scolastico include tre scuole: scuola dell'infanzia, scuola primaria e
4	scuola secondaria inferiore. Ciascuna scuola è distribuita in diversi plessi e ciascun
5	plesso ospita anche più di una scuola.
6	Le iscrizioni avvengono nel mese di gennaio: i genitori iscrivono i propri figli e di
7	ognuno di loro viene registrata l'anagrafica.
8	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad una delle 3 scuole
9	ma si vuole comunque memorizzare lo storico delle sue iscrizioni.
10	Inoltre, all'atto dell'iscrizione, ogni studente è associato ad una classe.
11	Invece, di ciascun insegnante si conosce, ogni anno, il numero di ore lavorative
12	settimanali per ogni classe a cui è assegnato.
13	I plessi sono identificati da un indirizzo e sono composti da aule, di cui si conoscono
14	i metri quadri, il numero massimo di studenti ospitabili e la loro fascia di età.
15	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero (1-4 per
16	l'infanzia, 1-5 per la primaria e 1-3 per la secondaria inferiore) e da una lettera
17	dell'alfabeto (aule di gradi diversi possono avere la stessa coppia).

2.2.3. RAFFINAMENTO DELLE SPECIFICHE E INDIVIDUAZIONE DEI CONCETTI DI BASE

	FRASI DI CARATTERE GENERALE
	Il complesso scolastico StudioOnline vuole riorganizzare il proprio database a seguito di alcuni aggiornamenti nella normativa.

	FRASI RELATIVE A SCUOLA
	Il complesso scolastico include tre scuole: scuola dell'infanzia, scuola primaria e scuola secondaria inferiore. Ciascuna scuola è distribuita in diversi plessi e ciascun plesso ospita anche più di una scuola.

	FRASI RELATIVE ALL'ISCRIZIONE
	Le iscrizioni avvengono nel mese di gennaio dove per ciascuno studente viene registrata l'anagrafica.

	FRASI RELATIVE ALLO STUDENTE
	Ogni studente, in un anno scolastico (01/09-19/06), è iscritto ad uno dei 3 gradi dell'istituto ma si vuole comunque memorizzare lo storico delle sue iscrizioni. Inoltre, all'atto dell'iscrizione, ogni studente è associato ad una classe.

	FRASI RELATIVE ALL'INSEGNANTE
	Invece, di ciascun insegnante si conosce, ogni anno, il numero di ore lavorative settimanali per ogni classe a cui è assegnato.

	FRASI RELATIVE AL PLESSO
	Ciascuna scuola è distribuita in diversi plessi e ciascun plesso ospita anche più di una scuola. I plessi sono identificati da un indirizzo e sono composti da aule, di cui si conoscono i metri quadri, il numero massimo di studenti ospitabili e la loro fascia di età.

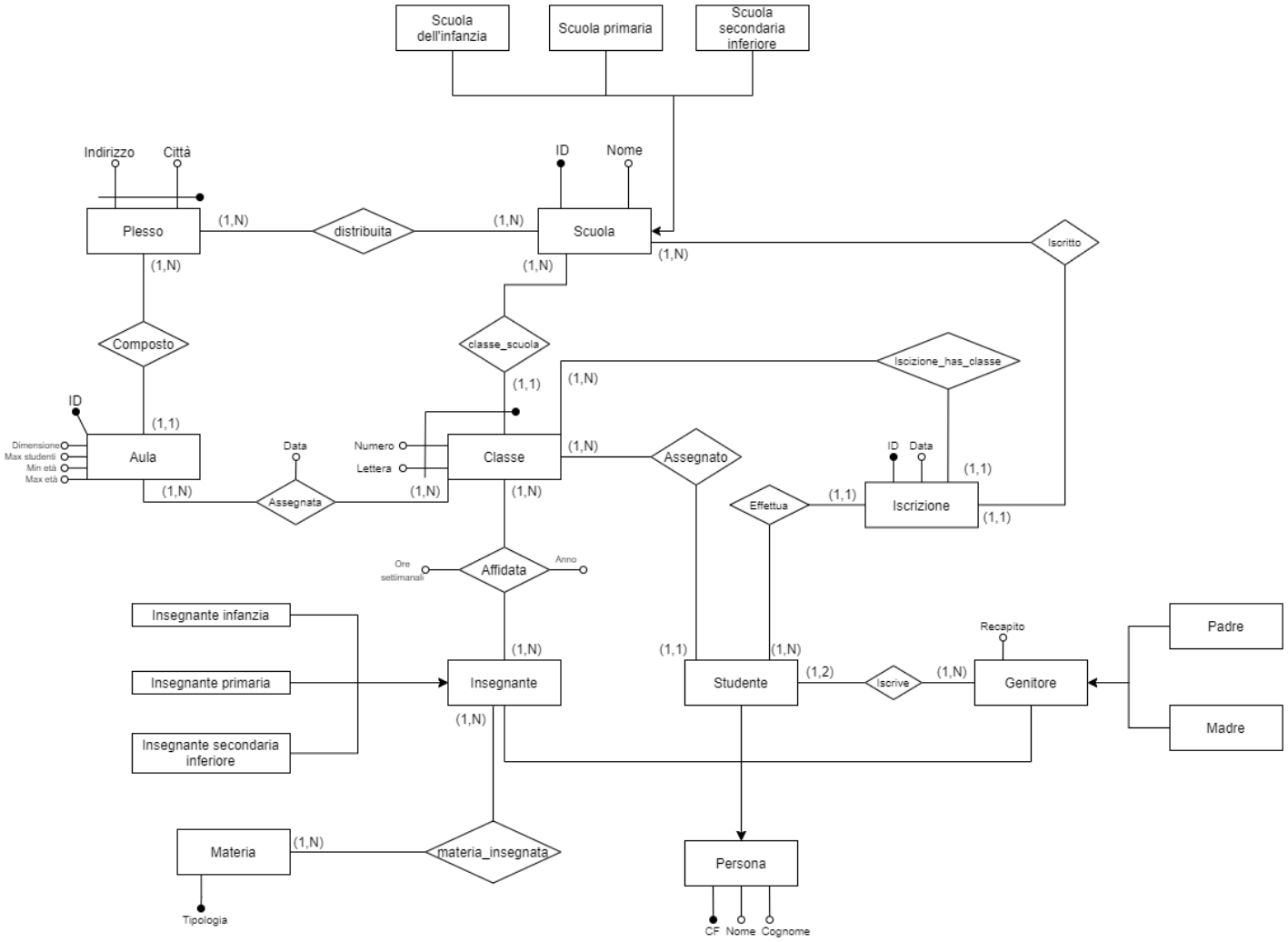
	FRASI RELATIVE ALLA CLASSE
	Ogni classe, che ogni anno può cambiare aula, è identificata da un numero (1-4 per l'infanzia, 1-5 per la primaria e 1-3 per la secondaria inferiore) e da una lettera dell'alfabeto (aule di gradi diversi possono avere la stessa coppia).

2.3. SPECIFICA DELLE OPERAZIONI SUI DATI PREVISTE

1	Inserire una nuova iscrizione.
2	Trovare gli studenti iscritti alla scuola secondaria inferiore nell'anno 2020.
3	Trovare gli studenti che hanno frequentato almeno una classe per ogni scuola dell'istituto.
4	Trovare i plessi con il numero di studenti ospitabili più alto.
5	Trovare, per ogni anno, le scuole con il maggior numero di iscritti.
6	Controllare che ad un'aula non vengano assegnati più iscritti di quanti ne possa ospitare.
7	Alla scadenza delle iscrizioni annuali (1 febbraio) si memorizzi il numero di nuovi iscritti per scuola e classe (1-4 infanzia, 1-5 primaria e 1-3 medie).

3. TEMATICA PROGETTUALE

3.1 SCHEMI E-R



3.2 DOCUMENTAZIONE DELLO SCHEMA E-R

3.2.1 DIZIONARIO DEI DATI

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Plesso	Struttura fisica di una determinate scuola.	Indirizzo,città	Indirizzo, città
Aula	Luogo in cui si svolgono le lezioni.	ID, dimensione, max_studenti, min_età, max_età	ID
Scuola	Tipo di istituizione scolastica.	ID, nome	ID
Studente	Persona iscritta ad una scuola.	CF, nome, cognome	CF

3.2.2 DESCRIZIONE ENTITÀ

PLESSO			
Descrizione	Struttura fisica di una determinate scuola.		
Nome Attributo	**Tipo di dato**	**Vincolo**	**Descrizione**
Indirizzo	stringa	Obbligatorio	Indirizzo del plesso
Città	stringa	Obbligatorio	Città in cui è situato un determinato plesso

STUDENTE			
Descrizione	Persona iscritta ad una scuola.		
Nome Attributo	**Tipo di dato**	**Vincolo**	**Descrizione**
CF	stringa	Obbligatorio	Codice identificativo dello Studente
nome	stringa	Obbligatorio	Nome dello studente
cognome	stringa	Obbligatorio	Cognome dello studente

SCUOLA			
Descrizione	Tipo di istituizione scolastica.		
Nome Attributo	**Tipo di dato**	**Vincolo**	**Descrizione**
ID	numerico	Obbligatorio	Numerico identificativo della scuola
nome	stringa	Obbligatorio	Nome della scuola

AULA			
Descrizione	Luogo in cui si svolgono le lezioni.		
Nome Attributo	**Tipo di dato**	**Vincolo**	**Descrizione**
ID	numerico	obbligatorio	Id identificativo dell'aula.
dimensione	numerico	Obbligatorio,maggiore di 0	Dimensione in metri quadri dell'aula
max_studenti	numerico	Obbligatorio,maggiore di 0	Numero massimo di studenti ospitabili
Min_età	numerico	Obbligatorio,maggiore di 0	Età minima degli studenti
Max_età	numerico	Obbligatorio,maggiore di 0	Età massima degli studenti

3.2.3 DESCRIZIONE RELAZIONI

RELAZIONE DISTRIBUITA	
DESCRIZIONE	DISTRIBUZIONE DELLE SCUOLE IN VARI PLESSI.
ENTITÀ COINVOLTE	
ENTITÀ	**CARDINALITÀ**
PLESSO	(1,N)
SCUOLA	(1,N)

RELAZIONE COMPOSTO	
DESCRIZIONE	AULE CHE COMPONGONO UN PLESSO.
ENTITÀ COINVOLTE	
ENTITÀ	**CARDINALITÀ**
PLESSO	(1,N)
AULA	(1,1)

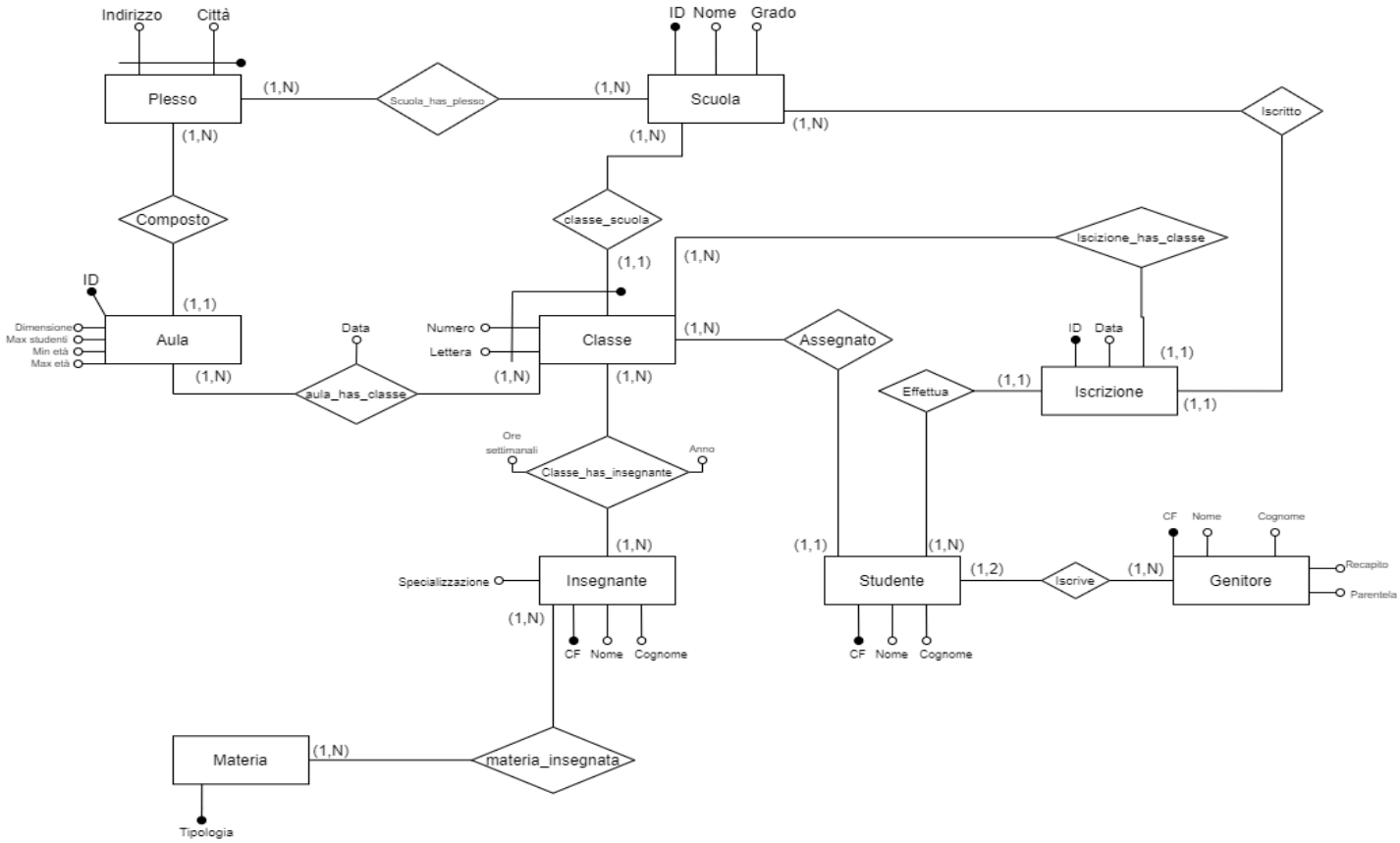
RELAZIONE ASSEGNATA		
DESCRIZIONE	CLASSE CHE PUÒ ESSERE ASSSEGNATA AD UN AULA IN UN DATO ANNO.	
ENTITÀ COINVOLTE		
ENTITÀ	CARDINALITÀ	
AULA	(1,N)	
CLASSE	(1,N)	
ATTRIBUTI		
NOME	TIPO DI DATO	DESCRIZIONE
DATA	NUMERICO	DATA DI ASSEGNAMENTO DI UNA CLASSE AD UN'AULA.

3.2.4 VINCOLI NON ESPRESSI DALLO SCHEMA E-R

REGOLE DI VINCOLO
1) Il numero della classe deve essere compreso tra 1-4 per l'infanzia, 1-5 per la primaria, 1-3 per la secondaria inferiore. 2) Lo studente non può iscriversi dopo il giorno 1 febbraio.

4. PROGETTAZIONE LOGICA

4.1 RISTRUTTURAZIONE DELLO SCHEMA E-R



4.2 DOCUMENTAZIONE DELLO SCHEMA LOGICO

4.2.1 DESCRIZIONE ENTITÀ

Entità Plesso			
Descrizione	Struttura fisica di una determinate scuola.		
Nome Attributo	**Tipo di Dato**	**Vincolo**	**Descrizione**
indirizzo	Varchar (50)	Primary key	Indirizzo in cui è situato il plesso.
città	Varchar (30)	Primary key	Città in cui è situato il plesso.

Entità Scuola			
Descrizione	Tipo di istituizione scolastica.		
Nome Attributo	**Tipo di Dato**	**Vincolo**	**Descrizione**
Id	int	Primary key	Id identificativo della scuola.
Nome	Varchar (50)	Not null	Nome della scuola
Tipo	Varchar (30)	Not null	Indica I diversi tipi di scuola .

Entità Aula			
Descrizione	Luogo in cui si svolgono le lezioni.		
Nome Attributo	**Tipo di Dato**	**Vincolo**	**Descrizione**
Id	int	Primary key	Id identificativo dell'aula.
Dimensione	Unsigned int	Not null	Dimensione in metri quadri dell'aula.
Max_studenti	Tinyint unsigned	Not null	Massimo numero di studenti ospitabili .
Min_età	Tinyint unsigned	Not null	Minima età degli studenti ospitabili.
Max_età	Tinyint unsigned	Not null	Massima età degli studenti ospitabili.

STUDENTE			
Descrizione	Persona iscritta ad una scuola.		
Nome Attributo	**Tipo di dato**	**Vincolo**	**Descrizione**
Cf	Char (16)	Primary key	Codice identificativo dello Studente
nome	Varchar (30)	Not null	Nome dello studente
cognome	Varchar (30)	Not null	Cognome dello studente

4.2.2 DESCRIZIONE RELAZIONI

RELAZIONE DISTRIBUITA	
DESCRIZIONE	DISTRIBUZIONE DELLE SCUOLE IN VARI PLESSI.
ENTITÀ COINVOLTE	
ENTITÀ	**CARDINALITÀ**
PLESSO	(1,N)
SCUOLA	(1,N)

RELAZIONE COMPOSTO	
DESCRIZIONE	AULE CHE COMPONGONO UN PLESSO.
ENTITÀ COINVOLTE	
ENTITÀ	**CARDINALITÀ**
PLESSO	(1,N)
AULA	(1,1)

RELAZIONE ASSEGNATA		
DESCRIZIONE	CLASSE CHE PUÒ ESSERE ASSSEGNATA AD UN AULA IN UN DATO ANNO.	
ENTITÀ COINVOLTE		
ENTITÀ	**CARDINALITÀ**	
AULA	(1,N)	
CLASSE	(1,N)	
ATTRIBUTI		
NOME	**TIPO DI DATI**	**DESCRIZIONE**
DATA	DATE	DATA DI ASSEGNAMENTO DI UNA CLASSE AD UN'AULA.

4.3 TRADUZIONE ENTITÀ

Aula(<u>ID</u>, dimensione, max_studenti,min_eta,max_eta, (citta_plesso, indirizzo_plesso)*)

Con vincolo di integrità referenziale tra gli attributi citta_plesso, indizizzo_plesso e la relazione plesso.

classe(<u>numero,lettera,scuola)</u>

Genitore(<u>CF</u>, nome, congome, parentela, recapito)

Insegnante(<u>CF,</u> nome, cognome, specializzazione)

Materia(<u>tipologia</u>)

Iscrizione(<u>ID</u>, data_iscrizione, studente*, (numero_classe, lettera_classe, scuola_classe)*)

Con vincolo di integrità referenziale tra gli attributi numero_classe, lettera_classe,scuola_classe e la relazione classe. Con vincolo di integrità referenziale tra l'attributo studente e la relazione studente.

Plesso(<u>indirizzo,citta)</u>

Scuola(<u>ID</u>, nome, grado)

Studente(<u>ID</u>, nome, cognome, genitore1*, genitore2, (lettera_classe, scuola_classe,numero_classe)*)

Con vincolo di integrità referenziale tra gli attributi numero_classe, lettera_classe,scuola_classe e la relazione classe. Con vincolo di integrità referenziale tra l'attributo genitore1 e la relazione genitore.

4.4 TRADUZIONE RELAZIONI

Aula_has_classe(data_assegnamento, (numero_classe,lettera_classe,scuola_classe)*, aula*)

Con vincolo di integrità referenziale tra gli attributi numero_classe, lettera_classe,scuola_classe e la relazione classe. Con vincolo di integrità referenziale tra l'attributo aula e la relazione aula.

Classe_has_insegnante(ore_settimanali, anno, (numero_classe, lettera_classe, scuola_classe)*, insegnante)

Con vincolo di integrità referenziale tra gli attributi numero_classe, lettera_classe,scuola_classe e la relazione classe. Con vincolo di integrità referenziale tra l'attributo insegnante e la relazione insegnante.

Scuola_has_plesso((citta_plesso,indirizzo_plesso)*,scuola*)

Con vincolo di integrità referenziale tra gli attributi citta_plesso, indirizzo_plesso e la relazione plesso. Con vincolo di integrità referenziale tra l'attributo scuola e la relazione scuola.

Materia_insegnata (materia*, insegnante*)

Con vincolo di integrità referenziale tra l'attributo materia e la relazione materia. Con vincolo di integrità referenziale tra l'attributo insegnante e la relazione insegnante.

5. PROGETTAZIONE FISICA

5.1 DEFINIZIONE DELLO SCHEMA DELLA BASE DI DATI

TABLE:

```
create table if not exists scuola
(
    id int auto_increment primary key,
    nome varchar(50) not null,
    tipo varchar(30) not null check (tipo = 'scuola infanzia' or tipo = 'scuola primaria' or tipo = 'scuola secondaria inferiore')
);

create table if not exists genitore
(
    cf char(16) primary key,
    nome varchar(30) not null,
    cognome varchar(30) not null,
    parentela char(5) not null check ( parentela = 'madre' or parentela = 'padre'),
    recapito char(10) not null
);

create table if not exists plesso
(
    indirizzo varchar(50),
    citta varchar(30),
    primary key(indirizzo, citta)
);

create table if not exists scuola_has_plesso
(
    citta_plesso varchar(30),
    indirizzo_plesso varchar(50),
    scuola int not null,
    foreign key (indirizzo_plesso, citta_plesso) references plesso(indirizzo,citta),
    foreign key (scuola) references scuola(id)
);
```

```sql
create table if not exists aula
(
    id int auto_increment primary key,
    dimensione int unsigned not null,
    max_studenti tinyint unsigned not null,
    min_eta tinyint unsigned not null,
    max_eta tinyint unsigned not null,
    citta_plesso varchar(30),
    indirizzo_plesso varchar(50),
    foreign key (indirizzo_plesso, citta_plesso) references plesso(indirizzo,citta)
);

create table if not exists classe
(
    numero tinyint unsigned,
    lettera char(1),
    scuola int,
    primary key (numero,lettera,scuola),
    foreign key(scuola) references scuola(id)
);

create table if not exists studente
(
    cf char(16) primary key,
    nome varchar(30) not null,
    cognome varchar(30) not null,
    scuola int not null,
    genitore1 char(16) not null,
    genitore2 char(16),
    lettera_classe char(1) not null,
    scuola_classe int not null,
    numero_classe tinyint unsigned not null,
    foreign key (numero_classe, lettera_classe, scuola_classe) references classe(numero,lettera,scuola),
    foreign key (scuola) references scuola(id),
    foreign key (genitore1) references genitore(cf)
);
create table if not exists iscrizione
(
    id int auto_increment primary key,
    data_iscrizione date not null,
    studente char(16) not null,
    numero_classe tinyint unsigned,
    lettera_classe char(1),
    scuola_classe int,
    foreign key (numero_classe,lettera_classe, scuola_classe) references classe(numero,lettera,scuola),
    foreign key (studente) references studente(cf)
);

create table if not exists aula_has_classe
(
    data_assegnamento date not null,
    numero_classe tinyint unsigned not null,
    lettera_classe char(1) not null,
    scuola_classe int not null,
    aula int not null,
    foreign key (aula) references aula(id),
    foreign key (numero_classe, lettera_classe, scuola_classe) references classe(numero,lettera,scuola)
);
create table if not exists insegnante
(
    cf char(16) primary key,
    nome varchar(30) not null,
    cognome varchar(30) not null,
    specializzazione varchar(30) not null
    check (specializzazione = 'insegnante infanzia' or specializzazione = 'insegnante primaria'
    or specializzazione = 'insegnante secondaria inferiore')
);
```

```sql
create table if not exists classe_has_insegnante
(
    ore_settimanali tinyint unsigned not null,
    anno date not null,
    numero_classe tinyint unsigned not null,
    lettera_classe char(1) not null,
    scuola_classe int not null,
    insegnante char(16) not null,
    foreign key (insegnante) references insegnante(cf),
    foreign key (numero_classe, lettera_classe, scuola_classe) references classe(numero,lettera,scuola)
);
create table if not exists materia
(
    tipologia varchar(30) primary key
);

create table if not exists materia_insegnata
(
    materia varchar(30),
    insegnante char(16),
    foreign key (materia) references materia(tipologia),
    foreign key (insegnante) references insegnante(cf)
);
```

TRIGGER:

```sql
delimiter //
create trigger controllo_periodo_iscrizione before insert on studente
for each row
begin
    if month(current_date()) >= 2 then
        signal sqlstate '45000' set message_text = 'Iscrizioni chiuse';
    end if;
end; //
delimiter ;

delimiter //
create trigger controllo_periodo_iscrizione_on_iscrizione before insert on iscrizione
for each row
begin
    if month(current_date()) >= 2 then
        signal sqlstate '45000' set message_text = 'Iscrizioni chiuse';
    end if;
end; //
delimiter ;

delimiter $$
create trigger iscrivi_studente after insert on studente
for each row
begin
    insert into iscrizione(data_iscrizione, studente, numero_classe, lettera_classe, scuola_classe)
    values (current_date(), new.cf,new.numero_classe, new.lettera_classe, new.scuola_classe);
end; $$
delimiter ;
```

```
delimiter $$
create trigger controllo_numero_classe before insert on classe
for each row
begin

    set @tipo = (select distinct tipo from scuola inner join classe on scuola.id = classe.scuola where scuola.id = new.scuola);

    if @tipo = 'scuola infanzia' and (new.numero < 1 or new.numero > 4) then
        signal sqlstate '45000' set message_text = 'Il numero della classe inserita non è valido per il tipo di scuola';

    elseif @tipo = 'scuola primaria' and (new.numero < 1 or new.numero > 5) then
        signal sqlstate '45000' set message_text = 'Il numero della classe inserita non è valido per il tipo di scuola';

    elseif  @tipo = 'scuola secondaria inferiore' and (new.numero < 1 or new.numero > 3) then
        signal sqlstate '45000' set message_text = 'Il numero della classe inserita non è valido per il tipo di scuola';

    end if;
end; $$
delimiter ;

delimiter $$
create trigger vietato_assembramento before insert on studente
for each row
begin
    set @num_studenti_attuali = (select count(*)
                                 from classe
                                 inner join studente on studente.numero_classe = classe.numero
                                 and studente.lettera_classe = classe.lettera
                                 where classe.scuola = new.scuola);

    set @max_studenti = (select max_studenti from aula inner join aula_has_classe on aula_has_classe.aula = aula.id
                         where aula_has_classe.lettera_classe = new.lettera_classe
                         and aula_has_classe.numero_classe = new.numero_classe);
    if @max_studenti < @num_studenti_attuali + 1 then
        signal sqlstate '45000' set message_text = 'Troppi studenti nell\'aula';
    end if;
end; $$
delimiter ;
```

```
delimiter !!
create trigger controllo_professore_classe before insert on classe_has_insegnante
for each row
begin
    set @specializzazione_insegnante = (select specializzazione from insegnante where cf = new.insegnante);
    set @scuola = (select tipo from scuola where id = new.scuola_classe);

    if @specializzazione_insegnante = 'insegnante infanzia' and @scuola = 'scuola primaria' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

    elseif @specializzazione_insegnante = 'insegnante infanzia' and @scuola = 'scuola secondaria inferiore' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

    elseif @specializzazione_insegnante = 'insegnante primaria' and @scuola = 'scuola infanzia' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

    elseif @specializzazione_insegnante = 'insegnante primaria' and @scuola = 'scuola secondaria inferiore' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

    elseif @specializzazione_insegnante = 'insegnante secondaria inferiore' and @scuola = 'scuola infanzia' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

     elseif @specializzazione_insegnante = 'insegnante secondaria inferiore' and @scuola = 'scuola primaria' then
        signal sqlstate '45000' set message_text = 'Non puoi insegnare in questa classe';

    end if;
end; !!
delimiter ;
```

EVENT:

```
delimiter //
create event if not exists studenti_iscritti
#((now() - interval(dayofyear(now()) - 1) day) + interval 1 year ) + interval 1 month
on schedule every 1 year starts '2021-02-01 00:00:00'
do
begin
    drop view if exists totale_iscritti;
    create view totale_iscritti as
        select tipo,numero_classe, lettera_classe, count(*) as totale_iscritti
        from scuola
        inner join iscrizione on iscrizione.scuola_classe = scuola.id
        group by scuola_classe, numero_classe, lettera_classe, year(data_iscrizione);
end; //
delimiter ;
```

5.2 DEFINIZIONE DELLE INTERROGAZIONI PER LA VISUALIZZAZIONE DEI DATI

5.2.1 SCRIPT SQL – DML: ANALISI PUNTUALI

```
# Inserire una nuova iscrizione
insert into iscrizione(data_iscrizione, studente, numero_classe, lettera_classe, scuola_classe) values (now(), 'spsfnc98s12d005a', 3, 'B', 1)
```

```
#trovare gli studenti iscritti alla scuola secondaria inferiore nell'anno 2020
select cf, nome, cognome from studente
inner join iscrizione on iscrizione.studente = studente.cf
where year(iscrizione.data_iscrizione) = '2020'
and (select  distinct tipo
    from scuola
    inner join (
    select *
    from iscrizione
    inner join classe on iscrizione.scuola_classe = classe.scuola) as iscrizione_classe
    on iscrizione_classe.scuola_classe = scuola.id
) = 'scuola secondaria inferiore'

# Trovare gli studenti che hanno frequentato almeno una classe per ogni grado dell'istituto.
select cf, nome, cognome, genitore1 from studente
where cf in
    (select cf from (select distinct *
                    from scuola
                    inner join (
                                    select studente.cf, iscrizione.scuola_classe, iscrizione.data_iscrizione
                                    from studente
                                    inner join iscrizione on iscrizione.studente = studente.cf
                                ) as tuple on tuple.scuola_classe = scuola.id
                ) as tabella
            group by cf
            having count(*) >=3
    )
```

5.2.2 SCRIPT SQL – DML: ANALISI AGGREGATE

```
# Trovare i plessi con il numero di studenti ospitabili più alto.
select citta_plesso, indirizzo_plesso, max(somma)
from ( select citta_plesso, indirizzo_plesso , sum(max_studenti) as somma
       from aula
       group by citta_plesso,indirizzo_plesso
     ) as somma_aula

#Trovare, per ogni anno, le scuole con il maggior numero di iscritti.
select scuola_classe, max(studenti_iscritti)
from(
select scuola_classe, count(studente) as studenti_iscritti
from iscrizione
group by year(data_iscrizione), scuola_classe) as tabella
```