

Schema Logico

1. ALUNNO(CF, nome, cognome, data_nascita, sostegno)
2. INSEGNANTE(CF, nome, cognome, data_nascita, di_ruolo, materia)
3. CLASSE(id, numero, lettera, id_scuola*)
4. AULA(id, mq, #max:stud, nome_fasciaEtà*, piano, id_plesso*)
5. PLESSO(id, indirizzo, #piani, ascensore)
6. FASCIAETÀ(nome, min, max)
7. SCUOLA(id, nome, telefono, tipo, tempoPieno)
8. INSEGNA(id_ins*, id_classe*, AS, ore)
9. ASSEGNAMENTO(id_classe*, id_aula*, AS, numerosa)
10. STORICOISCRIZIONE(id_alunno*, id_classe*, AS, data)

con vincoli di integrità

1. Un'aula non può trovarsi ad un piano superiore al numero di piani di un plesso
2. Non può esistere un alunno con lo stesso CF di un insegnante
3. Un'aula A non può avere numero massimo di studenti superiore a quello di un'altra aula B se i mq di A sono inferiori ai mq di B
4. In un anno scolastico alla stessa aula può essere assegnata solo una classe
5. In un anno scolastico uno studente può iscriversi solo ad una classe
6. Ogni classe deve avere un numero di studenti non inferiore a 15 e minore di 25, in un dato anno scolastico
7. Non esistono due classi con la stessa tripla <numero, lettera, scuola>
8. Ogni plesso con più di due piani deve avere un ascensore
9. Un insegnante non può insegnare in più di due classi nello stesso anno scolastico
10. Ogni classe deve avere almeno un insegnante di ruolo in un dato anno scolastico
11. In ogni classe non possono esserci più di 2 insegnanti di ruolo in un dato anno scolastico

ASSERZIONI

A1: Un'aula non può trovarsi ad un piano superiore al numero di piani di un plesso

Create Assertion PianoAula(

```
    check(not exists(select *
                      from Aula, Plesso
                      where plesso=Plesso.id and piano>#piani))
```

);

A2: Non può esistere un alunno con lo stesso CF di un insegnante

Create Assertion CfAlunnoInsegnante(

```
    check(not exists(select *
                      from Alunno, Insegnante
                      where Alunno.CF=Insegnante.CF))
```

);

A3: Un'aula A non può avere numero massimo di studenti superiore a quello di un'altra aula B se i mq di A sono inferiori ai mq di B

```
Create Assertion MqAule(  
    check(not exists(select *  
        from Aula A1,Aula A2  
        where A1.mq<A2.mq and A1.#max:stud>A2.#max:stud))  
);
```

A4: In un anno scolastico alla stessa aula può essere assegnata solo una classe

```
Create Assertion UnicaClasse(  
    check(not exists(select*  
        from Assegnamento A1, Assegnamento A2  
        where A1.AS=A2.AS and A1.id_aula=A2.id_aula  
        and A1.id_classe <>A2.id_classe))  
);
```

A5: in un anno scolastico uno studente può iscriversi solo ad una classe

```
Create Assertion UnicaIscrizione(  
    check(not exists(select*  
        from StoricoIscrizione S1, StoricoIscrizione S2  
        where S1.AS=S2.AS and S1.id_alunno=S2.id_alunno and  
        S1.id_classe<>S2.id_classe))  
);
```

A6: Ogni classe, in un dato anno scolastico, deve avere un numero di studenti non inferiore a 15 e minore di 25

```
Create Assertion ClasseMinima(  
    check(not exists(Select id_classe From StoricoIscrizione GroupBy id_class, AS  
        having count(*) < 15 and count(*) >= 25)  
    )
```

oppure

```
Create Assertion ClasseMinima(  
    CHECK (15 <= ALL (SELECT count(*)  
        FROM StoricoIscrizione  
        GROUPBY id_class, AS  
    and  
    25> ALL (SELECT count(*)  
        FROM StoricoIscrizione  
        GROUPBY id_class, AS)  
    )
```

A7: Non esistono due classi con la stessa tripla <numero, lettera, scuola>

```
Create Assertion ClasseId(  
    check(not exists(select *  
        from Classe A1, Classe A2  
        where A1.numero=A2.numero and A1.lettera=A2.lettera and A1.scuola=A2.scuola and A1.id_classe <>A2.id_classe))
```

```

        check(not exists(select *
                        from Classe C1,Classe C2
                        where C1.id <>C2.id and C1.numero=C2.numero and
                        C1.lettera=C2.lettera and C1.scuola=C2.scuola))
    );

```

A8: Ogni plesso con più di due piani deve avere un ascensore

```

Create Assertion PlessoAscensore(
    check(not exists(select *
                    from Plesso
                    where #piani>2 and ascensore=false) )
)

```

A9: Un insegnante non può insegnare in più di due classi nello stesso anno scolastico

```

Create Assertion Insegna(
    check(2 >= ALL (Select count(*) From Insegna GroupBy id_ins, AS))
)

```

A10: Ogni classe deve avere almeno un insegnante di ruolo in ogni anno scolastico

```

Create Assertion InsRuolo1(
    check(not exists(Select id_classe, AS
                    From Insegna Z
                    where not exists (
                        Select id_ins
                        From Insegna X, Insegnante Y
                        Where X.id_ins=Y.CF and Y.ruolo=true and
                        Z.id_classe=X.id_classe and Z.AS=X.AS ))
    )
)

```

NOTA: la seguente versione non funziona in quanto la condizione è sempre verificata

```

Create Assertion InsRuolo(
    check(1 <= ALL (Select count(*)
                    From Insegna X, Insegnante Y
                    Where X.id_ins=Y.CF and Y.ruolo=true
                    GroupBy id_classe, AS))
)

```

A11: In ogni classe non possono esserci più di 2 insegnanti di ruolo in un dato anno scolastico

```

Create Assertion InsRuolo2(
    check(3 > ALL (Select count(*)
                    From Insegna X, Insegnante Y
                    Where X.id_ins=Y.CF and Y.ruolo=true
                    GroupBy id_classe, AS))
)

```