

Esame di stato 2004 Portfolio studente

Traccia

Il Dirigente Scolastico di una Scuola Secondaria Superiore chiede che si realizzi una base di dati per l'archiviazione e la gestione di informazioni riguardanti le attività scolastiche ed extrascolastiche documentate, nonché i risultati scolastici conseguiti da ciascuno studente al fine di produrre, in itinere e/o al termine del corso di studi, un "portfolio studente".

L'organizzazione scolastica dell'istituto prevede che:

- a) ciascuno studente possa frequentare più corsi di recupero e/o di sostegno e/o di arricchimento dell'offerta formativa;
- b) ogni corso abbia un titolo, una descrizione, una data di inizio e di fine, un monte ore definito;
- c) ogni studente possa frequentare più corsi esterni alla scuola;
- d) i corsi esterni alla scuola hanno un titolo, una descrizione, una data di inizio e di fine, un monte ore definito, un riferimento che indichi l'Ente e/o l'Istituzione che li ha organizzati, un riferimento alla documentazione di accertamento;
- e) ogni classe sia individuata univocamente da un numero ordinale progressivo (I, II, III, IV, V) e da una lettera che ne indica la sezione di appartenenza (A, B, C, D, E, F, G,).

In particolare, il Dirigente Scolastico chiede che si possa procedere all'archiviazione dei:

- dati anagrafici degli studenti utili alla loro univoca identificazione;
- dati relativi alla frequenza delle classi del corso di studi (quali classi ciascuno studente ha frequentato in ordine crescente ed in quale anno scolastico);
- dati relativi agli esiti conclusivi di ciascun anno scolastico (promozione sì/no, eventuali debiti formativi ed in quale disciplina) per ciascuno studente;
- dati relativi ai corsi interni;
- dati relativi ai corsi esterni alla scuola purché documentati.

Il candidato, fatte le opportune ipotesi aggiuntive, progetti una base di dati utile alla realizzazione

del portfolio studente richiesto dal Dirigente Scolastico, fornendo:

1. uno schema concettuale della base di dati;
2. uno schema logico della base di dati;
3. la definizione delle relazioni della base di dati in linguaggio SQL;

ed inoltre:

implementi in linguaggio SQL le seguenti interrogazioni:

- data una classe, quali studenti di quella classe hanno frequentato corsi e di che tipo;
- dato uno studente, quali corsi ha frequentato, di che tipo, per quale monte ore e in quale anno scolastico;
- dato un anno scolastico, quali corsi interni sono stati attivati e da quali studenti sono stati seguiti;
- dato un corso, quali sono i dati relativi ad esso e per quali anni scolastici è stato attivato;
- dato uno studente quali classi ha frequentato, in quali anni scolastici e con quali esiti finali.

Analisi

Per l'analisi della realtà di interesse si utilizza la modellazione Entità-Relazioni che produce un diagramma ER.

Individuazione delle entità.

Entità 'studente': è l'anagrafica dei studenti. Ogni studente deve essere registrato al primo accesso alla scuola in una entità di tipo anagrafico. La registrazione è permanente, anche dopo l'uscita dello studente dalla scuola, a causa delle registrazioni di tipo storico che lo coinvolgono. L'esigenza di identificare in modo univoco ogni studente porta alla definizione delle seguenti proprietà indispensabili: *cognome, nome, sesso, data di nascita, comune di nascita, stato di nascita*. Questo insieme definisce una chiave candidata (codice fiscale, che può anche essere calcolato automaticamente a partire dai dati). Per i nati in paesi stranieri non è definito il comune di nascita e il codice fiscale contiene invece la nazione di nascita. Per motivi di efficienza si inserisce una chiave primaria artificiale ad autoincremento e il codice fiscale, che può essere derivato, non viene inserito.

Ulteriori informazioni utili per la gestione degli studenti, ma non indispensabili, sono: *indirizzo, telefono, comune di residenza, e cittadinanza* che può non coincidere con la nazionalità di nascita a causa di successive variazioni (naturalizzazione).

Gli attributi comune di nascita, stato di nascita, comune di residenza e cittadinanza possono essere sottoposti a normalizzazione determinando le tabelle di lookup comune e stato associate 1:N all'entità studente.

I campi sono tutti obbligatori eccetto comune di nascita che è nullo in caso di stato di nascita straniero e telefono.

Entità 'comune': è una tabella di look-up contenente il catalogo dei comuni italiani (circa 8000) Ogni comune è identificato univocamente da una chiave artificiale a codice parlante definita da norme nazionali. Il codice è alfanumerico formato da un carattere alfabetico dalla 'A' alla 'Z' seguito da una stringa numerica di tre cifre. Ad ogni codice è associata una descrizione. I comuni sono ordinati alfabeticamente per codice.

I campi sono tutti obbligatori

Entità 'stato': è una tabella di look-up contenente il catalogo degli stati riconosciuto dallo stato italiano (circa 300) Ogni stato è identificato univocamente da una chiave artificiale a codice parlante definita da norme nazionali. Il codice è alfanumerico formato da un carattere alfabetico dalla 'A' alla 'Z' seguito da una stringa numerica di tre cifre.

I campi sono tutti obbligatori

Entità 'classe': è un catalogo delle classi che possono essere attivate in ogni anno scolastico nella scuola. La registrazione è permanente, indipendentemente dalla effettiva attivazione della classe in un particolare anno scolastico a causa delle registrazioni di tipo storico che lo coinvolgono. Una possibile chiave candidata è la composizione del numero progressivo della classe combinato con la sezione. Tuttavia per ottenere una soluzione generale si introduce una chiave primaria ad autoincremento mentre la chiave candidata diventa un attributo composto in cui ogni campo è unico ed indicizzato. I campi sono tutti obbligatori.

Entità corso': è un catalogo dei corsi che possono essere offerti dalla scuola agli studenti. I corsi si dividono in interni ed esterni attraverso una gerarchia ISA. Elementi comuni ad entrambe le categorie sono la chiave primaria artificiale ad autoincremento, la descrizione ed il monte ore. I corsi interni hanno inoltre l'attributo di tipo di corso che per motivi di normalizzazione viene associato all'entità "tipo". I corsi esterni hanno, oltre agli attributi comuni, l'associazione all'entità "ente" e l'attributo "tipo di certificazione". In mancanza di ulteriori informazioni sul significato di tale attributo si sceglie un tipo testo. La registrazione è permanente, indipendentemente dalla effettiva attivazione di un corso in un particolare anno scolastico a causa delle registrazioni di tipo storico che lo coinvolgono.

I campi sono tutti obbligatori eccetto "tipo di certificazione"

Entità 'tipo': è un catalogo dei tipi di corso interno che possono essere offerti dalla scuola agli studenti. Svolge le funzioni di lookup nei confronti dell'entità corso garantendo la normalizzazione della banca dati. Ogni tipo di corso è identificato da una chiave primaria ad autoincremento e da una descrizione di tipo testo. La registrazione è permanente, indipendentemente dalla effettiva attivazione di un tipo corso in un particolare anno scolastico a causa delle registrazioni di tipo storico che lo coinvolgono.
I campi sono tutti obbligatori.

Entità 'ente': è un catalogo degli enti abilitati a fornire corsi esterni agli studenti. Svolge le funzioni di lookup nei confronti dell'entità corso garantendo la normalizzazione della banca dati. Ogni ente è identificato da una chiave primaria ad autoincremento e da una descrizione di tipo testo. La registrazione è permanente, indipendentemente dalla effettiva attivazione di un tipo corso in un particolare anno scolastico a causa delle registrazioni di tipo storico che lo coinvolgono.
I campi sono tutti obbligatori.

Entità 'anno_scolastico': è un catalogo degli anni scolastici gestiti dall'archivio storico della banca dati. Ogni anno scolastico è identificato univocamente da una chiave artificiale ad autoincremento e da un campo di testo che contiene la descrizione dell'anno scolastico.
I campi sono tutti obbligatori

Entità 'esito': è un catalogo dei possibili esiti delle attività scolastiche. Non tutte le attività scolastiche hanno un esito quindi l'associazione a questa entità è facoltativa ma sicuramente esiste per la frequenza alle classi mentre si può ipotizzare che esista anche per i corsi di recupero. Ogni esito è identificato univocamente da una chiave artificiale ad autoincremento e da un campo di testo che contiene la descrizione dell'esito come 'promosso', 'non promosso' 'trasferito' ...
I campi sono tutti obbligatori

Entità 'disciplina': è un catalogo di tutte le discipline insegnate nella scuola e che possono essere oggetto di debito. Ogni disciplina è identificata univocamente da una chiave artificiale ad autoincremento e da un campo di testo che contiene la descrizione della disciplina. I campi sono tutti obbligatori

Individuazione delle associazioni.

Entità debole 'classe attivata': In ogni anno scolastico, in base alle nuove iscrizioni, alle promozioni, non promozioni ed ai trasferimenti in entrata, la scuola attiva un certo numero di classi. L'attivazione di una classe è registrata nell'entità debole 'classe_attivata'. Questa entità è identificata da una chiave primaria artificiale ad autoincremento e dall'associazione alla tipologia di classe (entità classe) e all'anno scolastico di attivazione (entità anno_scolastico). A questa entità vengono associate le frequenze degli studenti.

Entità debole 'corso attivato': In ogni anno scolastico, in base all'offerta formativa la scuola attiva un certo numero di corsi interni e riconosce l'effettuazione di un certo numero di corsi esterni organizzati da altri enti. L'attivazione di un corso interno o esterno è registrata nell'entità debole 'corso_attivato'. Questa entità è identificata da una chiave primaria artificiale ad autoincremento e dall'associazione alla tipologia di corso (entità corso) e all'anno scolastico di attivazione (entità anno_scolastico) la data di inizio e di fine del corso. A questa entità vengono associate le frequenze degli studenti.

Entità debole 'frequenza': In ogni anno scolastico, gli studenti che frequentano sono iscritti ad una classe attivata in base alla prima iscrizione, promozione, non promozione o trasferimento in entrata. Inoltre gli studenti che sono iscritti ad una classe possono facoltativamente frequentare altri corsi interni o esterni. Sia l'iscrizione alla classe sia l'iscrizione ai corsi facoltativi è registrata nell'entità debole 'frequenza' che associa uno studente alla classe attivata o al corso attivato che frequenta.

Questa entità è strutturata in modo gerarchico ISA identificata da un chiave primaria artificiale ad autoincremento comune e di una associazione comune a studente. Le due entità che la compongono sono frequenza_classe e frequenza_corso. La prima è dotata dell'associazione 'con esito' e dell'associazione 'alla classe' mentre la seconda è dotata dell'associazione 'al corso'.

Associazione studente/comune 'nato a': ogni studente di nazionalità italiana deve essere associato ad un comune di nascita selezionato dall'elenco di lookup dei comuni italiani. Esiste quindi una associazione 1:N tra studente e comune. La relazione è parziale da entrambi i lati (un comune può non essere associato ad alcuno studente, uno studente non è associato ad alcun comune se è straniero)

Associazione studente/comune 'risiede': ogni studente deve essere associato ad un comune di residenza selezionato dall'elenco di lookup dei comuni italiani. Esiste quindi una associazione 1:N tra studente e comune. La relazione è parziale dal lato comune (un comune può non essere associato ad alcuno studente) e totale dal lato studente(uno studente deve sempre avere un comune di residenza)

Associazione studente/nazione 'nazionalità': ogni studente di nazionalità straniera deve essere associato ad una nazione di nascita selezionata dall'elenco di lookup degli stati riconosciuti dallo stato italiano. Esiste quindi una associazione 1:N tra studente e stato. La relazione è parziale da entrambi i lati (uno stato può non essere associato ad alcuno studente, uno studente non è associato ad alcuno stato se è italiano)

Associazione studente/nazione 'cittadinanza': ogni studente di cittadinanza diversa da quella italiana deve essere associato ad una nazione di cittadinanza selezionata dall'elenco di lookup degli stati riconosciuti dallo stato italiano. Esiste quindi una associazione 1:N tra studente e stato. La relazione è parziale da entrambi i lati (uno stato può non essere associato ad alcuno studente, uno studente non è associato ad alcuno stato se è italiano)

Associazione studente/frequenza 'ha frequentato': ogni studente presente nella scuola deve frequentare una classe mentre gli studenti degli anni precedenti sono archiviati storicamente in classi del passato. Inoltre uno studente può frequentare o avere frequentato in passato corsi facoltativi interni o esterni. L'informazione di frequenza è quindi una associazione 1:N tra frequenza e studente. La relazione è totale dal lato frequenza (una frequenza deve essere associata ad uno studente) e parziale dal lato studente (uno studente inserito in anagrafica potrebbe non avere alcuna frequenza)

Associazione freq_classe/classe attivata 'alla classe': ogni frequenza di classe deve appartenere ad una classe attivata. In questo modo lo studente è indirettamente associata ad una classe attivata in un particolare anno scolastico. Esiste quindi una associazione 1:N tra freq_classe e classe attivata. La relazione è totale da entrambi i lati

Associazione freq_corso/corso attivato 'al corso': ogni frequenza di corso deve appartenere ad un corso attivato. In questo modo lo studente è indirettamente associato ad un corso attivato in un particolare anno scolastico. Esiste quindi una associazione 1:N tra freq_corso e corso attivato. La relazione è totale da entrambi i lati

Associazione freq_classe/esito 'con esito': ogni frequenza di classe è associata ad un esito della frequenza. Esiste quindi una associazione 1:N tra freq_classe e esito. La relazione è totale da entrambi i lati

Associazione freq_classe/disciplina 'debit': ogni frequenza di classe può essere associata a debiti in nessuna, una o più discipline. Esiste quindi una associazione N:N tra freq_classe e discipline. La relazione è parziale da entrambi i lati

Associazione classe_attivata/classe 'attivazione cl': ogni classe effettivamente attivata in un certo anno scolastico appartiene ad una tipologia di classe. Esiste quindi una associazione 1:N tra classe attivata e classe. La relazione è parziale dal lato classe (una classe può non essere attivata in un certo anno scolastico) e totale dal lato classe attivata(una classe attivata deve appartenere ad una tipologia di classe prevista)

Associazione corso attivato/corso 'attivazione cor': ogni corso effettivamente attivato in un certo anno scolastico appartiene ad una tipologia di corso. Esiste quindi una associazione 1:N tra corso attivato e corso. La relazione è parziale dal lato corso (una corso può non essere attivata in un certo anno scolastico) e totale dal lato corso attivato(un corso attivato deve appartenere ad una tipologia di corso prevista)

Associazione classe attivata/anno 'att cl anno': associa una classe effettivamente attivata all'anno di attivazione.

Associazione corso attivato/anno 'att cor anno': associa un corso effettivamente attivato all'anno di attivazione.

Associazione interno/tipo tipo interno': associa un corso interno alla sua tipologia

Associazione esterno/ente 'eff da': associa un corso esterno al suo ente erogatore

Ipotesi aggiuntive

Anagrafe studenti: per l'identificazione dello studente si inseriscono i dati del codice fiscale e di i dati di residenza. Per garantire l'integrità delle informazioni comune di nascita, di residenza, stato di nascite e cittadinanza sono normalizzati in tabelle di lookup ottenute secondo le normative nazionali.

Archivio storico: Il problema chiede di potere estrarre informazioni di tipo storico quindi tutte le registrazioni effettuate devono essere mantenute con una associazione all'anno di registrazione.

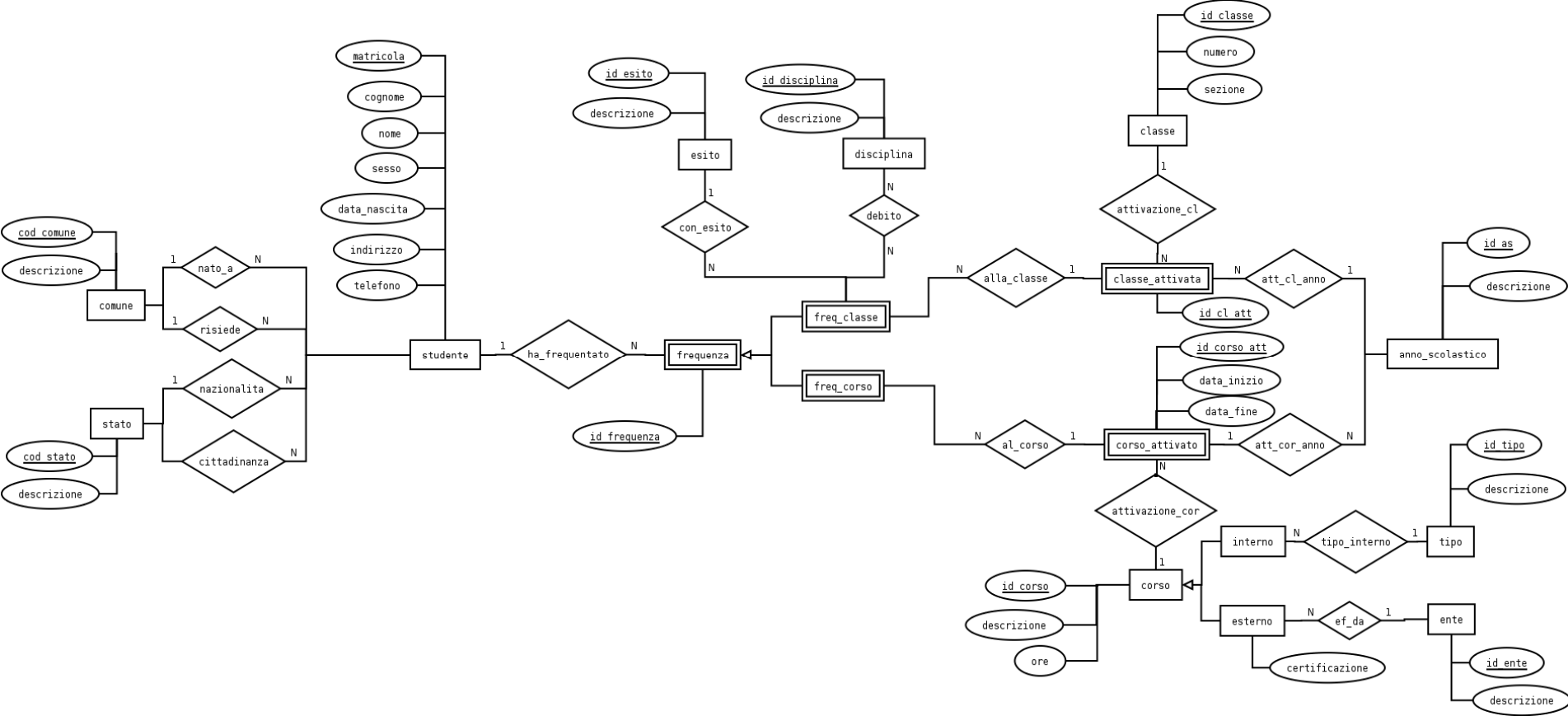
Frequenza delle classi: Si ipotizza che la frequenza ad una classe sia obbligatoria da parte di uno studente iscritto quindi ogni studente ha almeno una registrazione storica di frequenza non necessariamente nell'anno in corso.

Frequenza dei corsi: Si ipotizza che la frequenza ad un corso interno/esterno sia facoltativa quindi uno studente può non avere alcuna registrazione storica di frequenza ad alcun tipo di corso.

Esito delle frequenze: l'esito delle frequenze viene esteso da un semplice booleano ad una lookup table che tiene conto anche di altre situazioni diverse dal promosso/non promosso come ritirato, trasferito, abbreviazione ...

Documentazione di accertamento dei corsi esterni: in mancanza di informazioni più dettagliate su questa proprietà si ipotizza di utilizzare un campo di testo libero nel quale può essere inserita una informazione riguardante la certificazione del corso (numero del certificato, tipo di certificato ...)

Schema concettuale (modello ER)



Schema logico

Si usa il modello relazionale.

Ogni entità ed entità debole viene sostituita con una relazione.

Ogni associazione 1:N tra due entità viene sostituita con l'esportazione della chiave primaria del lato 1 come chiave esterna nel lato N.

Ogni associazione N:N tra due entità viene sostituita con la generazione di una nuova relazione ottenuta esportando le due chiavi primarie come chiavi esterne e definendo la loro composizione come chiave primaria della nuova relazione.

Poiché il progetto non richiede il modello fisico nello schema logico vengono anche indicati i tipi di dato.

Relazioni

studente(matricola, cognome, nome, sesso, data_n, indirizzo, telefono, comune, stato, comune, cittadinanza)

comune(cod_comune, descrizione)

stato(cod_stato, descrizione)

classe(id_classe, numero, sezione)

corso(id_corso, descrizione, ore, interno_esterno, certificazione, id_ente, id_tipo)

La gerarchia ISA viene collassata in una relazione unica. L'attributo discriminatore è il booleano interno_esterno (interno=1) mentre certificazione, id_ente ed id_tipo sono opzionali.

anno(id_anno, descrizione)

classe_att(id_classe_att, id_anno, id_classe)

corso_att(id_corso_att, id_anno, id_corso, data_i, data_f)

esito(id_esito, descrizione)

frequenza(id_frequenza, matricola, classe_corso, id_classe_att, id_corso_att, id_esito)

La gerarchia ISA viene collassata in una relazione unica. L'attributo discriminatore è il booleano classe_corso (classe=1) mentre id_classe_att, id_corso_att e id_esito sono opzionali.

disciplina(id_disciplina, descrizione)

debiti(id_frequenza, id_disciplina)

Vincoli di integrità

Vincoli di integrità referenziale

studente.comunen \subseteq comune.cod_comune

studente.comuner \subseteq comune.cod_comune

studente.staton \subseteq stato.cod_stato

studente.cittadinanza \subseteq stato.cod_stato

classe_att.id_classe \subseteq classe.id_classe

classe_att.id_anno \subseteq anno.id_anno

corso_att.id_corso \subseteq corso.id_corso

corso_att.id_anno \subseteq anno.id_anno

frequenza.matricola \subseteq studente.matricola

frequenza.id_classe_att \subseteq studente.id_classe_att | \emptyset

frequenza.id_corso_att \subseteq studente.id_corso_att | \emptyset

frequenza.id_esito \subseteq esito.id_esito | \emptyset

debito.id_frequenza \subseteq frequenza.id_frequenza

debito.id_debito \subseteq disciplina.id_disciplina

Vincoli esterni

Se interno_esterno=1 \rightarrow certificazione, ente=NULL; tipo NOT NULL

Se interno_esterno=0 \rightarrow certificazione, ente= NOT NULL; tipo NULL

Se classe_corso=1 \rightarrow id_corso_att=NULL; id_classe_att, id_esito NOT NULL

Se classe_corso=0 \rightarrow id_corso_att= NOT NULL; id_classe_att, id_esito NULL

Modello fisico

```
CREATE TABLE stato (  
  cod_stato VARCHAR(4) NOT NULL,  
  descrizione VARCHAR(80) NULL,  
  PRIMARY KEY(cod_stato)  
);  
  
CREATE TABLE comune (  
  cod_comune VARCHAR(4) NOT NULL,  
  descrizione VARCHAR(80) NULL,  
  PRIMARY KEY(cod_comune)  
);  
  
CREATE TABLE studente (  
  matricola INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  cognome VARCHAR(20) NOT NULL,  
  nome VARCHAR(20) NOT NULL,  
  sesso TINYINT(1) UNSIGNED NOT NULL,  
  datan DATE NOT NULL,  
  indirizzo VARCHAR(80) NOT NULL,  
  telefono VARCHAR(80) NULL,  
  PRIMARY KEY(matricola),  
  comunen VARCHAR(4) NOT NULL,  
  comuner VARCHAR(4) NOT NULL,  
  nazionalita VARCHAR(4) NOT NULL,  
  cittadinanza VARCHAR(4) NOT NULL,  
  INDEX idx_comunen(comunen),  
  INDEX idx_comuner(comuner),  
  INDEX idx_nazionalita(nazionalita),  
  INDEX idx_cittadinanza(cittadinanza),  
  FOREIGN KEY(comunen)  
    REFERENCES comune(cod_comune),  
  FOREIGN KEY(comuner)  
    REFERENCES comune(cod_comune),  
  FOREIGN KEY(nazionalita)  
    REFERENCES stato(cod_stato),  
  FOREIGN KEY(cittadinanza)  
    REFERENCES stato(cod_stato)  
);  
  
CREATE TABLE classe (  
  id_classe INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  numero VARCHAR(4) NOT NULL,  
  sezione VARCHAR(4) NOT NULL,  
  PRIMARY KEY(id_classe)  
);  
  
CREATE TABLE tipo (  
  id_tipo INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  descrizione VARCHAR(80) NOT NULL,  
  PRIMARY KEY(id_tipo)  
);  
  
CREATE TABLE ente (  
  id_ente INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  descrizione VARCHAR(80) NOT NULL,  
  PRIMARY KEY(id_ente)  
);
```

```

CREATE TABLE corso (
  id_corso INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descrizione VARCHAR(80) NOT NULL,
  ore INTEGER UNSIGNED NOT NULL,
  interno_esterno TINYINT(1) UNSIGNED NOT NULL DEFAULT 1,
  certificazione VARCHAR(255) NULL,
  id_tipo INTEGER UNSIGNED NOT NULL,
  id_ente INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(id_corso),
  INDEX idx_id_ente(id_ente),
  INDEX idx_id_tipo(id_tipo)
);

CREATE TABLE anno (
  id_anno INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descrizione VARCHAR(20) NOT NULL,
  PRIMARY KEY(id_anno)
);

CREATE TABLE classe_att (
  id_classe_att INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  id_anno INTEGER UNSIGNED NOT NULL,
  id_classe INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(id_classe_att),
  INDEX idx_id_anno(id_anno),
  INDEX idx_id_classe(id_classe),
  FOREIGN KEY(id_anno)
    REFERENCES anno(id_anno),
  FOREIGN KEY(id_classe)
    REFERENCES classe(id_classe)
);

CREATE TABLE corso_att (
  id_corso_att INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  id_anno INTEGER UNSIGNED NOT NULL,
  id_corso INTEGER UNSIGNED NOT NULL,
  datai DATE NOT NULL,
  dataf DATE NOT NULL,
  PRIMARY KEY(id_corso_att),
  INDEX idx_id_anno(id_anno),
  INDEX idx_id_corso(id_corso),
  FOREIGN KEY(id_anno)
    REFERENCES anno(id_anno),
  FOREIGN KEY(id_corso)
    REFERENCES corso(id_corso)
);

CREATE TABLE esito (
  id_esito INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descrizione VARCHAR(20) NOT NULL,
  PRIMARY KEY(id_esito)
);

```

```

CREATE TABLE frequenza (
  id_frequenza INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  matricola INTEGER UNSIGNED NOT NULL,
  classe_corso TINYINT(1) UNSIGNED NOT NULL DEFAULT 1,
  id_classe_att INTEGER UNSIGNED NULL,
  id_corso_att INTEGER UNSIGNED NULL,
  id_esito INTEGER UNSIGNED NULL,
  PRIMARY KEY(id_frequenza),
  INDEX idx_id_classe_att(id_classe_att),
  INDEX idx_id_corso_att(id_corso_att),
  INDEX idx_id_esito(id_esito),
);

CREATE TABLE disciplina (
  id_disciplina INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  descrizione VARCHAR(20) NOT NULL,
  PRIMARY KEY(id_disciplina)
);

CREATE TABLE debito (
  id_disciplina INTEGER UNSIGNED NOT NULL,
  id_frequenza INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(id_disciplina,id_frequenza),
  INDEX idx_id_disciplina(id_disciplina),
  INDEX idx_id_frequenza(id_frequenza),
  FOREIGN KEY(id_disciplina)
    REFERENCES disciplina(id_disciplina),
  FOREIGN KEY(id_frequenza)
    REFERENCES frequenza(id_frequenza)
);

```

Interrogazioni

a. data una classe, quali studenti di quella classe hanno frequentato corsi e di che tipo

Con lo schema realizzato è necessaria una ipotesi aggiuntiva: ci si riferisce ad una classe di un particolare anno scolastico e anche i corsi estratti si riferiscono allo stesso anno scolastico.

Si effettua una select annidata.

La parte interna proietta una tabella virtuale contenente i numeri di matricola degli studenti che hanno frequentato una particolare classe in un particolare anno forniti come parametri di ingresso.

La parte esterna proietta le informazioni su studenti iscritti a corsi effettuati nell'anno selezionato.

La nullità della chiave esterna id_corso_att in frequenza esclude le frequenze a classi dal natural join.

Il left join con tipo è necessario per includere anche i corsi esterni che avendo la chiave id_tipo nulla verrebbero esclusi.

```
SELECT
    t1.matricola,
    t1.cognome,
    t1.nome,
    t4.descrizione AS des_corso,
    t4.interno_esterno,
    t6.descrizione AS des_tipo_corso
FROM
    studente AS t1,
    frequenta AS t2,
    corso_att AS t3,
    corso AS t4,
    anno AS t5,
    left join tipo AS t6 on T4.id_tipo=t6.id_tipo
WHERE
    t1.matricola=t2.matricola
AND    t2.id_corso_att=t3.id_corso_att
AND    t3.id_corso=t4.id_corso
AND    t3.id_anno=t5.id_anno
AND    t3.id_anno='$anno'
AND    t1.matricola IN (
        SELECT
            t6.matricola
        FROM
            studente AS t6,
            frequenta AS t7,
            classe_att AS t8,
            classe AS t9,
            anno AS t10
        WHERE
            t6.matricola=t7.matricola
        AND    t7.id_classe_att=t8.id_classe_att
        AND    t8.id_classe=t9.id_classe
        AND    t8.id_anno=t10.id_anno
        AND    t9.numero='$numero'
        AND    t9.sezione='$sezione'
        AND    t10.descrizione='$anno'
    )
```

b. dato uno studente, quali corsi ha frequentato, di che tipo, per quale monte ore e in quale anno scolastico;

Si suppone di disporre del numero di matricola dello studente come parametro di ingresso.

La query proietta le informazioni sulle iscrizioni a corsi dello studente specificato in restrizione in tutti gli anni scolastici.

La nullità della chiave esterna id_corso_att in frequenza esclude le frequenze a classi dal natural join.

Il left join con tipo è necessario per includere anche i corsi esterni che avendo la chiave id_tipo nulla verrebbero esclusi.

L'ordinamento è per anno decrescente

```
SELECT
    t4.descrizione AS des_corso
    t4.interno_esterno
    t6.descrizione AS des_tipo_corso
    t4.ore,
    t5.descrizione AS des_anno
FROM
    studente AS t1,
    frequenta AS t2,
    corso_att AS t3,
    corso AS t4,
    anno AS t5,
    left join tipo AS t6 on T4.id_tipo=t6.id_tipo
WHERE
    t1.matricola=t2.matricola
AND    t2.id_corso_att=t3.id_corso_att
AND    t3.id_corso=t4.id_corso
AND    t3.id_anno=t5.id_anno
AND    t1.matricola = $matricola
ORDER BY
    des_anno DESC, interno_esterno,des_tipo_corso, des_corso
```

c. dato un anno scolastico, quali corsi interni sono stati attivati e da quali studenti sono stati seguiti;

Si suppone di disporre dell'anno scolastico richiesto come parametro di ingresso.

La query proietta le informazioni sulle iscrizioni a corsi dello studente specificato in restrizione in tutti gli anni scolastici.

La nullità della chiave esterna id_corso_att in frequenza esclude le frequenze a classi dal natural join.

La nullità della chiave esterna id_tipo in corso esclude i corsi esterni dal natural join.

L'ordinamento è gerarchico per tipo, corso, cognome, nome

```
SELECT
    t6.descrizione AS des_tipo_corso,
    t2.descrizione AS des_corso,
    t1.matricola,
    t1.cognome,
    t1.nome,
FROM
    studente AS t1,
    frequenta AS t2,
    corso_att AS t3,
    corso AS t4,
    anno AS t5,
    tipo AS t6
WHERE
    t1.matricola=t2.matricola
AND    t2.id_corso_att=t3.id_corso_att
AND    t3.id_corso=t4.id_corso
AND    T3.id_anno=t5.id_anno
AND    T4.id_tipo=t6.id_tipo
AND    t5.descrizione='$anno'
ORDER BY
    des_tipo_corso, des_corso, cognome, nome
```

d. dato un corso, quali sono i dati relativi ad esso e per quali anni scolastici è stato attivato;

Si suppone di disporre della chiave primaria di corso come parametro di ingresso.

La query proietta le informazioni sul corso e sulle sue attivazioni

La nullità della chiave esterna id_corso_att in frequenza esclude le frequenze a classi dal natural join.

Il left join con tipo è necessario per includere anche i corsi esterni che avendo la chiave id_tipo nulla verrebbero esclusi.

L'ordinamento è per anno decrescente

```
SELECT
    t5.descrizione AS des_anno,
    t3.descrizione AS des_corso,
    t3.data_inizio
    t3.data_fine
    t3.ore,
    t6.descrizione AS des_tipo_corso,
    t7.descrizione AS des_ente_corso,
    t7.certificazione
FROM
    corso_att AS t3,
    corso AS t4,
    anno AS t5,
    left join tipo AS t6 on t4.id_tipo=t6.id_tipo
    left join ente AS t7 on t4.id_ente=t7.id_ente
WHERE
    t3.id_corso=t4.id_corso
AND    t3.id_anno=t5.id_anno
AND    t4.id_corso=$id_corso
ORDER BY
    des_anno DESC
```

e. dato uno studente quali classi ha frequentato, in quali anni scolastici e con quali esiti finali.

Questa query è la duale della precedente query b. Anche in questo caso si suppone di disporre del numero di matricola dello studente come parametro di ingresso.

La query proietta le informazioni sulle iscrizioni a classi dello studente specificato in restrizione in tutti gli anni scolastici.

La nullità della chiave esterna `id_classe_att` in frequenza esclude le frequenze a corsi facoltativi.

Poiché non è esplicitamente richiesto non vengono elencati i debiti.

L'ordinamento è per anno crescente

```
SELECT
    t5.descrizione AS des_anno
    CONCAT(t4.numero,t4.sezione) as des_classe,
    descrizione AS des_corso,
    t6.descrizione AS des_esito
FROM
    studente AS t1,
    frequenta AS t2,
    classe_att AS t3,
    classe AS t4,
    anno AS t5,
    esito AS t6
WHERE
    t1.matricola=t2.matricola
AND
    t2.id_classe_att=t3.id_classe_att
AND
    t3.id_classe=t4.id_classe
AND
    t3.id_anno=t5.id_anno
AND
    t3.id_esito=t6.id_esito
AND
    t1.matricola = $matricola
ORDER BY
    des_anno ASC
```