

ESERCITAZIONI: II PROVA SCRITTA DI INFORMATICA

(1) Una casa editrice desidera archiviare in un database le informazioni riguardanti gli **abbonamenti** alle **riviste** ed ai **giornali** pubblicati tra il 1995 ed il 2006

Per ogni **abbonato** si richiede di memorizzare i dati anagrafici, per ogni **abbonamento** la data ed il periodo di validità (trimestrale, semestrale, annuale). Bisogna considerare che gli abbonati possono avere abbonamenti anche per più pubblicazioni.

Per ogni giornale o rivista occorre archiviare il titolo, la periodicità (quotidiano, settimanale, mensile, il prezzo dell'abbonamento e gli argomenti trattati. Inoltre deve essere mantenuto un indice con i titoli dei principali articoli pubblicati ed a ciascun articolo deve essere associata la pubblicazione in cui è comparso.

Si realizzino, fatte le ipotesi aggiuntive del caso,

- Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:

Q1: Dato il titolo di una pubblicazione, ricercare gli articoli pubblicati in un determinato anno;

Q2: Dato il titolo di una pubblicazione, ricercare gli abbonati annuali;

Q3: Dato il nominativo di un abbonato, stabilire a quante riviste è abbonato;

Q4: Dato un argomento, elencare le pubblicazioni in cui è trattato;

Q5: Riportare per ogni pubblicazione il numero di abbonamenti;

Q6: Visualizzare i giornali con almeno 5000 abbonati annuali;

Q7: Dati i titoli di due pubblicazioni, visualizzarne gli abbonati comuni;

Q8: Dato il titolo di una pubblicazione, elencare le pubblicazioni che trattano i suoi stessi argomenti.

Svolgimento

Premessa: Una **base di dati** (in inglese **database**) può essere considerata come una raccolta di dati progettati per essere fruiti in maniera ottimizzata da differenti applicazioni e da utenti diversi.

Una base di dati, per poter essere definita tale, deve essere:

- **sicura:** ossia deve essere progettata in modo da impedire che essa possa essere danneggiata da eventi accidentali (come crash di sistema) o da accessi non autorizzati;

- **integra:** ossia deve essere garantito che le operazioni effettuate da utenti autorizzati non possano provocare una perdita di consistenza dei dati;

- **consistente:** ossia i dati in essa contenuti devono essere significativi ed effettivamente utilizzabili nelle applicazioni dell'azienda per cui è stata progettata;

- **condivisibile:** ossia applicazioni ed utenti diversi devono poter accedere, secondo opportune modalità, ai dati comuni;

- **persistente:** ossia deve avere un tempo di vita che non è limitato a quello delle singole esecuzioni dei programmi che la utilizzano (il contrario dei dati gestiti in memoria centrale)

- **efficiente:** l'utilizzo delle risorse deve essere ottimizzato riguardo ai ben noti parametri tempo (efficiente utilizzo della CPU) e spazio (efficiente uso della memoria).

L'insieme di attività che costituiscono l'attività di progettazione di una base di dati consta di tre distinte attività di progettazione:

1) **progettazione concettuale:** ha lo scopo di costruire e definire una rappresentazione corretta e completa della realtà di interesse concettuale in modo astratto ed indipendente dal DBMS che si intenderà utilizzare.

L'**input** di tale fase è il documento delle specifiche formali (nel nostro caso il testo dell'esercizio)

L'**output** di tale fase è uno schema concettuale ossia una rappresentazione astratta (indipendente dal DBMS) ed il più possibile formale della realtà (un esempio di output che vedremo in seguito è il diagramma E/R).

2) **progettazione logica**: ha lo scopo di trasformare lo schema concettuale (ancora astratto e indipendente da un DBMS) in uno schema logico ovvero in una rappresentazione efficiente rispetto alle strutture di un DBMS relativamente ad un ben definito modello dati (un esempio è una descrizione tramite tabelle del modello relazionale).

L'input di tale fase è il diagramma ER della fase di progettazione concettuale.

L'output di tale fase è uno schema logico riassumibile con relazioni rappresentate da tabelle logiche

3) **progettazione fisica**: ha lo scopo di implementare lo schema logico definendo tutti gli aspetti fisici di memorizzazione e rappresentazione in memoria di massa.

L'input di tale fase è lo schema logico (ossia le tabelle logiche) individuate nella fase di progettazione logica.

L'output di tale fase è l'implementazione in memoria di massa di tali tabelle (tabelle fisiche)

Da un'attenta lettura delle specifiche, si evidenzia che sono richieste le seguenti attività:

- gestione degli abbonati e dei loro abbonamenti;
- gestione delle pubblicazioni della casa editrice e degli argomenti trattati;
- raccolta degli articoli pubblicati

Più nel dettaglio volendo evidenziare quali sono le *specifiche sui dati* e quali le *specifiche sulle funzioni* possiamo ipotizzare che sarà necessario raccogliere informazioni relative:

- agli *abbonati* alle pubblicazioni;
- alle *pubblicazioni* della casa editrice (distinguendo tra *giornali* e *riviste*);
- agli *argomenti* trattati;
- agli *articoli* pubblicati.

Abbiamo così ottenuto un primo elenco di entità che dovranno entrare a far parte dello schema concettuale della base dati

A partire da questo elenco, associamo ad ogni entità individuata i corrispondenti attributi, attingendo sempre le informazioni dalle specifiche fornite:

- a ciascun abbonato deve essere associato un *Cognome*, un *Nome*, una *DataNascita*, un *Indirizzo*, un *Cap* ed una *Città* di residenza (anagrafica breve);
- ciascuna pubblicazione deve essere caratterizzata da un *Titolo*, dalla *Periodicità*, dal *Tipo* di pubblicazione (mapping di un'ISA con accorpamento delle entità figlie nell'entità padre – ISA totale), dal *Costo Trimestrale*, *Costo Semestrale*, *Costo Annuale* dell'abbonamento;
- ciascun argomento è caratterizzato da una sua *descrizione*;
- a ciascun articolo è associato un *titolo*, un *testo* ed un'eventuale *fotografia*.

Passiamo ora ad esaminare quali sono le associazioni tra le entità ipotizzate, individuando per ciascuna di esse la molteplicità e la totalità/parzialità in base alle caratteristiche di funzionalità evidenziate dalle specifiche nonché gli eventuali attributi:

- tra le entità *Abbonato* e *Pubblicazione* (entità padre dell'ISA) esiste un'associazione "*SiAbbona*" di molteplicità N:N, totale in entrambi i versi, in quanto "*un abbonato deve essere abbonato ad una o più pubblicazioni (supponendo che chi non rinnovi alcun abbonamento non sia più mantenuto nell'entità Abbonato) e viceversa una pubblicazione deve avere uno o più abbonati contemporaneamente*";
- tra le entità *Pubblicazione* ed *Argomento* esiste un'associazione "*Tratta*" di molteplicità N:N, totale in entrambi i versi, in quanto "*una pubblicazione deve trattare uno o più argomenti e viceversa un argomento deve essere trattato in una o più pubblicazioni*";
- tra le entità *Pubblicazione* ed *Articolo* esiste un'associazione "*Pubblica*" di molteplicità 1:N, totale in entrambi i versi, in quanto "*una pubblicazione deve pubblicare uno o più articoli e viceversa un articolo deve essere pubblicato in una pubblicazione*";

Sulla base della precedente analisi e delle specifiche rappresentiamo lo schema concettuale attraverso un diagramma E/R mostrato sotto.

Oltre alle specifiche fornite abbiamo introdotto alcune ipotesi aggiuntive:

(*) nello schema concettuale è stata individuata una *chiave primaria* per ogni entità;

(*) l'attributo *Fotografia* (opzionale) dell'entità *Articolo* conterrà il "path" completo di nome file dell'immagine che eventualmente accompagna il testo;

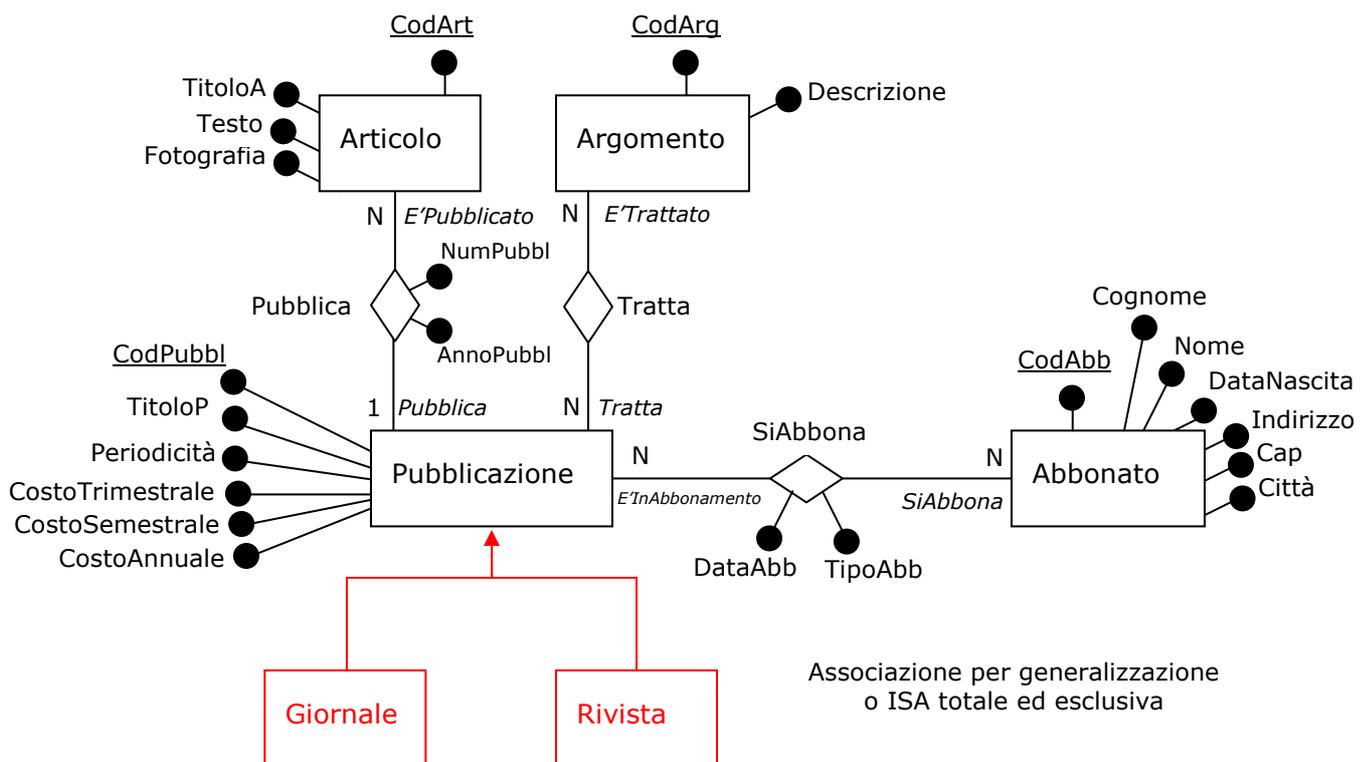
(*) gli attributi *DataAbb* e *TipoAbb* dell'associazione "*SiAbbona*" permettono di sapere per ogni abbonamento da quando è partito e di che tipo è (trimestrale, semestrale, annuale);

(*) gli attributi *NumPubbl* e *AnnoPubbl* dell'associazione "*Pubblica*" permettono di conoscere su quale numero della pubblicazione è stato pubblicato l'articolo ed in quale anno, in modo da realizzare l'*indice* degli articoli richiesto dalle specifiche;

Specifichiamo infine, oltre ai ben noti vincoli impliciti (rappresentati dai vincoli di chiave primaria e dalla totalità delle associazioni dirette e/o inverse) i seguenti vincoli espliciti:

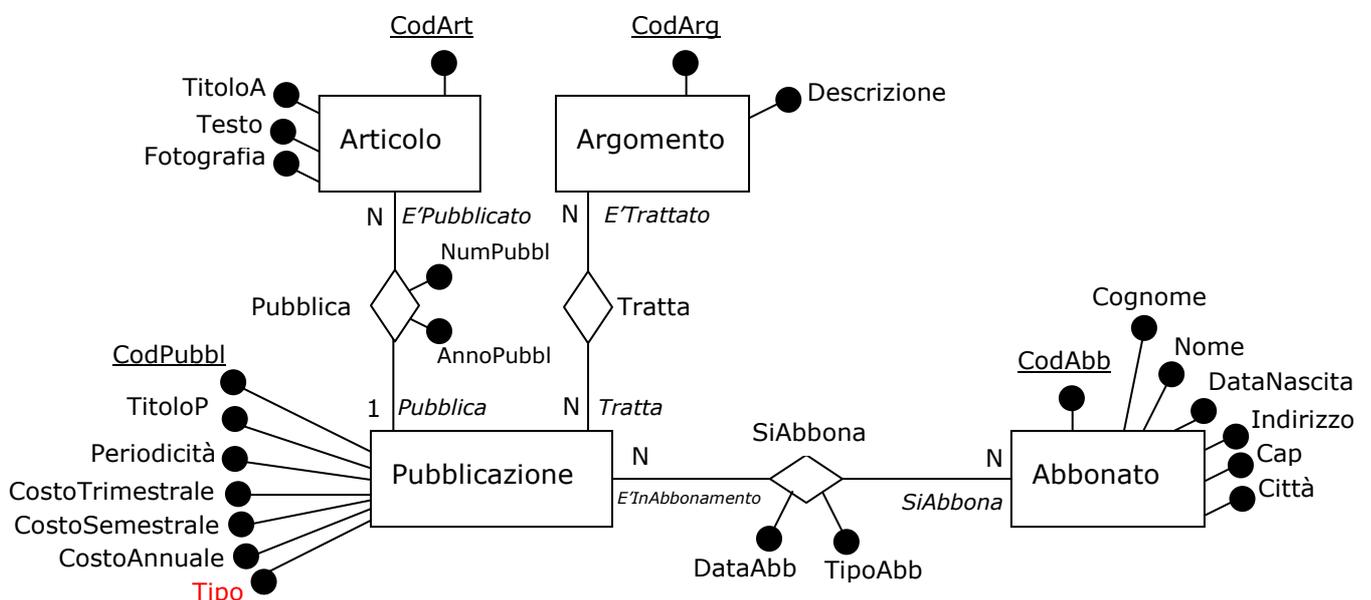
- il dominio dell'attributo *Periodicità* è composto dai valori "Quotidiano", "Settimanale" e "Mensile";
- il dominio dell'attributo *Tipo* è composto dai valori "Giornale" e "Rivista";
- il valore degli attributi *CostoTrimestrale*, *CostoSemestrale*, *CostoAnnuale* deve essere positivo
- il dominio dell'attributo *TipoAbb* è composto dai valori "Trimestrale", "Semestrale" e "Annuale";
- il valore dell'attributo *NumAbb* deve essere positivo e corrispondente ad una pubblicazione esistente;
- il valore dell'attributo *DataNascita* di un abbonato deve essere precedente rispetto al valore dell'attributo *DataAbb* del suo abbonamento.

Schema concettuale della realtà di interesse – diagramma E/R con ISA



(a) Schema concettuale della realtà di interesse – diagramma E/R senza ISA

(ottenuto attraverso l'accorpamento delle entità figlie nell'entità padre perché l'ISA è TOTALE)



Vincoli impliciti:

Sono quei vincoli direttamente deducibili dal diagramma E/R in quanto caratterizzati graficamente:

- vincoli di chiave primaria: tutti gli attributi sottolineati
- vincoli di integrità referenziale: totalità delle associazioni dirette e/o inverse (linea continua)

Vincoli espliciti:

Vincoli non deducibili direttamente dal diagramma E/R:

V1: (Pubblicazione.Periodicita = "Quotidiano") OR (Pubblicazione.Periodicita = "Settimanale") OR (Pubblicazione.Periodicita = "Mensile")

V2: (Pubblicazione.Tipo = "Giornale") OR (Pubblicazione.Tipo = "Rivista")

V3: (Pubblicazione.CostoTrimestrale > 0)

V4: (Pubblicazione.CostoSemestrale > 0)

V5: (Pubblicazione.CostoAnnuale > 0)

V6: (SiAbbona.TipoAbb = "Trimestrale") OR (SiAbbona.TipoAbb = "Semestrale") OR (SiAbbona.TipoAbb = "Annuale")

V7: (Pubblica.NumPubbl > 0)

V8: (Pubblica.DataPubbl <= "1995") AND (Pubblica.DataPubbl <= "2006")

V9: (Abbonato.DataNascita < SiAbbona.DataAbb)

(b) Schema logico della realtà di interesse - mapping relazionale del diagramma E/R

(i) *mapping dell'associazione "SiAbbona" di molteplicità N:N tra le entità "Abbonato" e "Pubblicazione"*

Pubblicazione (CodPubbl, TitoloP, Periodicita, CostoTrimestrale, CostoSemestrale, CostoAnnuale, Tipo)

Abbonato (CodAbb, Cognome, Nome, DataNascita, Indirizzo, Cap, Citta)

SiAbbona (CodAbb1, CodPubbl1, DataAbb, TipoAbb)

con "CodAbb1" chiave esterna (foreign key) sull'attributo "CodAbb" della relazione "Abbonato"

con "CodPubbl1" chiave esterna (foreign key) sull'attributo "CodPubbl" della relazione "Pubblicazione"

$VR_{\text{CodAbb1}}(\text{SiAbbona}) \subseteq VR_{\text{CodAbb}}(\text{Abbonato})$ dal mapping relazionale dell'associazione N:N

$VR_{\text{CodPubbl1}}(\text{SiAbbona}) \subseteq VR_{\text{CodPubbl}}(\text{Pubblicazione})$ dal mapping relazionale dell'associazione N:N

$VR_{\text{CodAbb}}(\text{Abbonato}) \subseteq VR_{\text{CodAbb1}}(\text{SiAbbona})$ dalla TOTALITA' dell'associazione diretta "SiAbbona"

$VR_{\text{CodPubbl}}(\text{Pubblicazione}) \subseteq VR_{\text{CodPubbl1}}(\text{SiAbbona})$ dalla TOTALITA' dell'associazione inversa "E'inAbbonamento"

(ii) *mapping dell'associazione "Tratta" di molteplicità N:N tra le entità "Pubblicazione" e "Argomento"*

Pubblicazione già mappata in precedenza

Argomento (CodArg, Descrizione)

Tratta (CodPubbl2, CodArg2)

con "CodPubbl2" chiave esterna (foreign key) sull'attributo "CodPubbl" della relazione "Pubblicazione"

con "CodArg2" chiave esterna (foreign key) sull'attributo "CodArg" della relazione "Argomento"

$VR_{\text{CodPubbl2}}(\text{Tratta}) \subseteq VR_{\text{CodPubbl}}(\text{Pubblicazione})$ dal mapping relazionale dell'associazione N:N

$VR_{\text{CodArg2}}(\text{Tratta}) \subseteq VR_{\text{CodArg}}(\text{Argomento})$ dal mapping relazionale dell'associazione N:N

$VR_{\text{CodPubbl}}(\text{Pubblicazione}) \subseteq VR_{\text{CodPubbl2}}(\text{Tratta})$ dalla TOTALITA' dell'associazione diretta "Tratta"

$VR_{\text{CodArg}}(\text{Argomento}) \subseteq VR_{\text{CodArg2}}(\text{Tratta})$ dalla TOTALITA' dell'associazione inversa "E'Trattato"

(iii) *mapping dell'associazione "Pubblica" di molteplicità 1:N tra le entità "Pubblicazione" e "Articolo"*

Pubblicazione già mappata in precedenza

Articolo (CodArt, TitoloA, Testo, Fotografia, NumPubbl, AnnoPubbl, CodPubbl3)

con "CodPubbl3" chiave esterna (foreign key) sull'attributo "CodPubbl" della relazione "Pubblicazione"

$VR_{\text{CodPubbl}}(\text{Pubblicazione}) \subseteq VR_{\text{CodPubbl3}}(\text{Articolo})$ dalla TOTALITA' dell'associazione diretta "Pubblica"

$VR_{\text{CodPubbl3}}(\text{Articolo}) \subseteq VR_{\text{CodPubbl}}(\text{Pubblicazione})$ dalla TOTALITA' dell'associazione inversa "E'Trattato"

Vincoli di integrità intrarelazionali (o interni) ed interrelazionali (o esterni)

I vincoli di chiave primaria (impliciti nel modello E/R) sono mappati in *vincoli intrarelazionali su più entuple*.

I vincoli di totalità di un'associazione (impliciti nel modello E/R) sono mappati in *vincoli interrelazionali di tipo referenziale*

I vincoli espliciti del diagramma E/R vengono mappati come segue:

V1 (Pubblicazione): "(Periodicità = "Quotidiano") OR (Periodicità = "Settimanale") OR (Periodicità = "Mensile)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V2 (Pubblicazione): "(Tipo = "Giornale") OR (Tipo = "Rivista)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V3 (Pubblicazione): "(CostoTrimestrale > 0)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V4 (Pubblicazione): "(CostoSemestrale > 0)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V5 (Pubblicazione): "(CostoAnnuale > 0)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V6 (SiAbbona): "(TipoAbb = "Trimestrale") OR (TipoAbb = "Semestrale") OR (TipoAbb = "Annuale)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V7 (Articolo): "(NumPubbl > 0)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V8 (Articolo): "(DataPubbl >= "1995") AND (DataPubbl <= "2006)" *vincolo intrarelazionale su singola entupla su singolo attributo*

V9(Abbonato, SiAbbona): "(Abbonato.DataNascita < SiAbbona.DataAbb)" *vincolo interrelazionale di tipo non referenziale*

Normalizzazione

La **teoria della normalizzazione** ha come scopo quello di fornire metodi per progettare basi di dati senza anomalie. La realtà che si intende analizzare tramite una base di dati può avere più schemi, equivalenti fra loro, che la descrivono. Alcuni di questi schemi possono presentare degli inconvenienti (**ridondanze o anomalie** di comportamento) che rendono inadeguato l'utilizzo efficiente delle informazioni.

Con **ridondanza** si intende la non necessaria ripetizione della stessa informazione con il conseguente spreco di spazio;

Con **anomalie da modifica** si intende la necessità di ripetere, in caso di cambiamento di una informazione, la modifica ovunque tale informazione è, duplicata;

Con **anomalia da inserimento** si intende la necessità di inserire ulteriori informazioni, non strettamente necessarie per poter inserire un nuovo dato;

Con **anomalie di cancellazione** si intende l'eliminazione di alcune informazioni in conseguenza della cancellazione di altre.

La teoria della normalizzazione si occupa di definire criteri di bontà per schemi relazionali (forme normali)

Una forma normale è una proprietà di uno schema relazionale che ne garantisce la "qualità", cioè l'assenza di determinati difetti) ed eventualmente di determinare metodi algoritmici per ottenere uno schema "migliore" ed equivalente a partire da uno schema non in forma normale (normalizzazione)

In genere per costruire uno schema relazionale si possono utilizzare i seguenti metodi:

1. Si parte da un buon schema E/R e lo si "traduce";
2. Si costruiscono direttamente le relazioni e poi si correggono quelle che presentano "anomalie";
3. Si analizza uno schema relazionale già esistente e lo si modifica/completa;

La normalizzazione

- È particolarmente utile se, per costruire lo schema, si usano i metodi (2) o (3);
- È moderatamente utile anche quando si usa il metodo (1);

PRIMA FORMA NORMALE: Ogni campo deve contenere un solo valore: (la 1NF è chiamata anche FORMA ATOMICA).

SECONDA FORMA NORMALE: Una tabella è in 2NF se è in 1NF e tutti gli attributi non chiave dipendono funzionalmente da tutta la chiave

TERZA FORMA NORMALE: Una tabella è in 3NF se è in 2NF e tutti gli attributi non chiave dipendono funzionalmente solo dalla chiave (eliminazione delle dipendenze funzionali transitive).

N.B. Il nostro schema relazionale è in 3FN

(c) Definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL

CREATE DATABASE CasaEditrice;

USE CasaEditrice;

CREATE DOMAIN MiaPeriodicita **AS** CHAR(11)

CHECK (VALUE IN ("Quotidiano", "Settimanale", "Mensile")); // Vincolo V1

CREATE DOMAIN MioTipo **AS** CHAR(8)

CHECK (VALUE IN ("Giornale", "Rivista")); // Vincolo V2

CREATE DOMAIN MioAbbonamento **AS** CHAR(11)

CHECK (VALUE IN ("Trimestrale", "Semestrale", "Annuale")); //Vincolo V6

CREATE TABLE Pubblicazione

```
(
CodPubbl          CHAR(4)          NOT NULL,
TitoloP           CHAR(25)         NOT NULL,
Periodicita       MiaPeriodicita   NOT NULL,
Costo Trimestrale DECIMAL(3,2)     NOT NULL,
Costo Semestrale  DECIMAL(3,2)     NOT NULL,
Costo Annuale     DECIMAL(3,2)     NOT NULL,
Tipo              MioTipo          NOT NULL,
PRIMARY KEY (CodPubbl),
UNIQUE (TitoloP),
CHECK (Costo Trimestrale > 0),           // Vincolo V3
CHECK (Costo Semestrale > 0),           // Vincolo V4
CHECK (Costo Annuale > 0)               // Vincolo V5
);
```

CREATE TABLE Abbonato

```
(
CodAbb           CHAR(10)         NOT NULL,
Cognome          CHAR(25)         NOT NULL,
Nome             CHAR(25)         NOT NULL,
DataNascita      DATE             NOT NULL,
Indirizzo        CHAR(100)        NOT NULL,
Cap              CHAR(5)          NOT NULL,
Citta            CHAR(25)         NOT NULL,
PRIMARY KEY (CodAbb),
UNIQUE (Cognome, Nome, DataNascita)
);
```

CREATE TABLE SiAbbona

```
(
CodAbb1          CHAR(10)         NOT NULL,
CodPubbl1        CHAR(4)          NOT NULL,
DataAbb          DATE             NOT NULL,
TipoAbb          MioAbbonamento  NOT NULL,
PRIMARY KEY (CodAbb1, CodPubbl1),
FOREIGN KEY (CodAbb1) REFERENCES ON Abbonato (CodAbb) // VR di chiave esterna
ON DELETE CASCADE, ON UPDATE CASCADE,
FOREIGN KEY (CodPubbl1) REFERENCES ON Pubblicazione (CodPubbl) // VR di chiave esterna
ON DELETE CASCADE, ON UPDATE CASCADE );
```

CREATE TABLE Argomento

```
(
CodArg          CHAR(4)          NOT NULL,
Descrizione     VARCHAR(500)     NOT NULL,
PRIMARY KEY (CodArg)
);
```

CREATE TABLE Tratta

```
(
CodPubbl2      CHAR(4)          NOT NULL,
CodArg2        CHAR(4)          NOT NULL,
PRIMARY KEY (CodPubbl2, CodArg2),
FOREIGN KEY (CodPubbl2) REFERENCES ON Pubblicazione (CodPubbl) // VR di chiave esterna
ON DELETE CASCADE, ON UPDATE CASCADE,
FOREIGN KEY (CodArg2) REFERENCES ON Argomento (CodArg) // VR di chiave esterna
ON DELETE CASCADE, ON UPDATE CASCADE
);
```

CREATE TABLE Articolo

```
(
CodArt          CHAR(4)          NOT NULL,
TitoloA         CHAR(25)         NOT NULL,
Testo           VARCHAR(2000)    NOT NULL,
Fotografia      CHAR(255),
NumPubbl       INT              NOT NULL,
AnnoPubbl      CHAR(4)          NOT NULL,
CodPubbl3      CHAR(4),
PRIMARY KEY (CodArt),
FOREIGN KEY (CodPubbl3) REFERENCES ON Pubblicazione (CodPubbl) // VR di chiave esterna
ON DELETE SET NULL, ON UPDATE CASCADE,
CHECK (NumPubbl > 0), // Vincolo V7
CHECK (AnnoPubbl BETWEEN "1995" AND "2006") // Vincolo V8
);
```

CREATE ASSERTION V9 CHECK (Abbonato.DataNascita < SiAbbona.DataAbb);

(d) Svolgimento delle query prima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL

Q1: Dato il titolo di una pubblicazione, ricercare gli articoli pubblicati in un determinato anno;

SQL dapprima occorre eseguire la connessione al database e poi

USE CasaEditrice;

```
SELECT TitoloA, Testo, Fotografia, NumPubbl
FROM Pubblicazione, Articolo
WHERE Pubblicazione.CodPubbl = Articolo.CodPubbl3
AND (AnnoPubbl = [InserisciAnno]) AND (TitoloP = [InserisciPubblicazione]);
```

N.B. e' possibile ordinarli per numero di pubblicazione

oppure anche

```
SELECT TitoloA, Testo, Fotografia, NumPubbl
FROM Pubblicazione INNER JOIN Articolo ON Pubblicazione.CodPubbl = Articolo.CodPubbl3
WHERE (AnnoPubbl = [InserisciAnno]) AND (TitoloP = [InserisciPubblicazione]);
```

Algebra Relazionale

$A = \{ \text{TitoloA, Testo, Fotografia, NumPubbl} \}$

$P = \{ (\text{AnnoPubbl} = [\text{InserisciAnno}]) \text{ AND } (\text{TitoloP} = [\text{InserisciPubblicazione}]) \}$

$Q1 = \Pi_A (\sigma_P (\text{Pubblicazione} \bowtie_{\text{CodPubbl}=\text{CodPubbl3}} \text{Articolo}))$

oppure anche in modo più efficiente

$Q1 = \Pi_A (\text{Pubblicazione} \bowtie_{\text{CodPubbl}=\text{CodPubbl3}} \sigma_P (\text{Articolo}))$

Q2: Dato il titolo di una pubblicazione, ricercare gli abbonati annuali;

SQL dapprima occorre eseguire la connessione al database e poi

USE CasaEditrice;

SELECT Cognome, Nome, DataNascita, Indirizzo, Cap, Citta

FROM Abbonato, SiAbbona, Pubblicazione

WHERE (Abbonato.CodAbb = SiAbbona.CodAbb1) **AND** (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl)

AND (TitoloP = [InserisciPubblicazione]);

oppure anche

N.B. e' possibile ordinarli per Cognome e Nome

SELECT Cognome, Nome, DataNascita, Indirizzo, Cap, Citta

FROM Abbonato **INNER JOIN** (SiAbbona **INNER JOIN** Pubblicazione **ON** CodPubbl1 = CodPubbl)

ON CodAbb = CodAbb1

WHERE (TitoloP = [InserisciPubblicazione]);

Algebra Relazionale

$A = \{ \text{Cognome, Nome, DataNascita, Indirizzo, Cap, Citta} \}$

$P = \{ (\text{TitoloP} = [\text{Inserisci Pubblicazione}]) \}$

$Q2 = \Pi_A (\sigma_P (\text{Abbonato} \bowtie_{\text{CodAbb}=\text{CodAbb1}} (\text{SiAbbona} \bowtie_{\text{CodPubbl1}=\text{CodPubbl}} \text{Pubblicazione})))$

oppure anche in modo più efficiente

$Q2 = \Pi_A (\text{Abbonato} \bowtie_{\text{CodAbb}=\text{CodAbb1}} (\text{SiAbbona} \bowtie_{\text{CodPubbl1}=\text{CodPubbl}} \sigma_P (\text{Pubblicazione})))$

Q3: Dato il nominativo di un abbonato, stabilire a quante riviste è abbonato

SQL dapprima occorre eseguire la connessione al database e poi

USE CasaEditrice;

SELECT COUNT(*) AS NumRiviste

FROM Abbonato, SiAbbona, Pubblicazione

WHERE (Abbonato.CodAbb = SiAbbona.CodAbb1) **AND** (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl)

AND (Cognome = [InserisciCognome]) **AND** (Nome = [InserisciNome]) **AND** (Tipo = "Rivista");

oppure anche

SELECT COUNT(*) AS NumRiviste

FROM Abbonato **INNER JOIN** (SiAbbona **INNER JOIN** Pubblicazione **ON** CodPubbl1 = CodPubbl)

ON CodAbb = CodAbb1

WHERE (Cognome = [InserisciCognome]) **AND** (Nome = [InserisciNome]) **AND** (Tipo = "Rivista");

Algebra Relazionale

$A = \{ \text{COUNT(*) AS NumRiviste} \}$ (N.B. un pò forzato quando vengono usate funzioni di aggregazione)

$P = \{ (\text{Cognome} = [\text{InserisciCognome}]) \text{ AND } (\text{Nome} = [\text{InserisciNome}]) \}$

$P1 = \{ (\text{Tipo} = \text{"Rivista"}) \}$

$Q3 = \Pi_A (\sigma_P (\text{Abbonato} \bowtie_{\text{CodAbb=CodAbb1}} (\text{SiAbbona} \bowtie_{\text{CodPubbl1=CodPubbl}} \text{Pubblicazione})))$

oppure anche in modo più efficiente

$Q3 = \Pi_A (\sigma_P (\text{Abbonato}) \bowtie_{\text{CodAbb=CodAbb1}} (\text{SiAbbona} \bowtie_{\text{CodPubbl1=CodPubbl}} \sigma_{P1} (\text{Pubblicazione})))$

Q4: Dato un argomento, elencare le pubblicazioni in cui è trattato

SQL dapprima occorre eseguire la connessione al database e poi

USE CasaEditrice;

SELECT TitoloP

FROM Pubblicazione, Tratta, Argomento

WHERE (Pubblicazione.CodPubbl = Tratta.CodPubbl2) **AND** (Tratta.CodArg2 = Argomento.CodArg)

AND (Descrizione = [InserisciDescrizione]);

oppure anche

N.B. E' possibile ordinarle per TitoloP

SELECT TitoloP

FROM Pubblicazione **INNER JOIN** (Tratta **INNER JOIN** Argomento **ON** CodArg2 =CodArg)

ON CodPubbl = CodPubbl2

WHERE (Descrizione = [InserisciDescrizione]);

Algebra Relazionale

$A = \{ \text{TitoloP} \}$

$P = \{ (\text{Descrizione} = [\text{InserisciDescrizione}]) \}$

$Q4 = \Pi_A (\sigma_P (\text{Pubblicazione} \bowtie_{\text{CodPubbl=CodPubbl2}} (\text{Tratta} \bowtie_{\text{CodArg2=CodArg}} \text{Argomento})))$

oppure anche in modo più efficiente

$Q4 = \Pi_A (\text{Pubblicazione} \bowtie_{\text{CodPubbl=CodPubbl2}} (\text{Tratta} \bowtie_{\text{CodArg2=CodArg}} \sigma_P (\text{Argomento})))$

Q5: Riportare per ogni pubblicazione il numero di abbonamenti

SQL dapprima occorre eseguire la connessione al database e poi

USE CasaEditrice;

SELECT CodPubbl, TitoloP, COUNT(*) AS NumAbbonati

FROM SiAbbona, Pubblicazione

WHERE (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl)

GROUP BY CodPubbl, TitoloP;

oppure anche

N.B. E' possibile ordinarle per TitoloP

SELECT CodPubbl, TitoloP, COUNT(*) AS NumAbbonati

FROM SiAbbona **INNER JOIN** Pubblicazione **ON** CodPubbl1 = CodPubbl

GROUP BY CodPubbl, TitoloP;

Q6: Visualizzare i giornali con almeno 5000 abbonati annuali

SQL dapprima occorre eseguire la connessione al database e poi
USE CasaEditrice;

N.B. E' possibile ordinarle per TitoloP

```
SELECT CodPubbl, TitoloP
FROM SiAbbona, Pubblicazione
WHERE (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl) AND (Tipo = "Giornale")
AND (TipoAbb = "Annuale")
GROUP BY CodPubbl, TitoloP
HAVING COUNT(*) >= 5000;
```

oppure anche

```
SELECT CodPubbl, TitoloP
FROM SiAbbona INNER JOIN Pubblicazione ON CodPubbl1 = CodPubbl)
WHERE (Tipo = "Giornale") AND (TipoAbb = "Annuale")
GROUP BY CodPubbl, TitoloP
HAVING COUNT(*) >= 5000;
```

Q7: Dati i titoli di due pubblicazioni, visualizzarne gli abbonati comuni

SQL dapprima occorre eseguire la connessione al database e poi
USE CasaEditrice;

```
(SELECT CodAbb, Cognome, Nome
FROM Abbonato, SiAbbona, Pubblicazione
WHERE (Abbonato.CodAbb = SiAbbona.CodAbb1) AND (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl)
AND (TitoloP = [InserisciPrimoTitolo]) )
```

INTERSECT

```
(SELECT CodAbb, Cognome, Nome
FROM Abbonato, SiAbbona, Pubblicazione
WHERE (Abbonato.CodAbb = SiAbbona.CodAbb1) AND (SiAbbona.CodPubbl1 = Pubblicazione.CodPubbl)
AND (TitoloP = [InserisciSecondoTitolo]) )
```

Q8: Dato il titolo di una pubblicazione, elencare le pubblicazioni che trattano i suoi stessi argomenti

SQL dapprima occorre eseguire la connessione al database e poi
USE CasaEditrice;

```
SELECT TitoloP
FROM Pubblicazione, Tratta
WHERE (Pubblicazione.CodPubbl = Tratta.CodPubbl2)
AND CodArg2 IN (SELECT CodArg
FROM Pubblicazione, Tratta
WHERE (Pubblicazione.CodPubbl = Tratta.CodPubbl2)
AND (TitoloP = [Inserisci Titolo]) ) ;
```

N.B. E' possibile ordinarle per TitoloP

(ESAME DI STATO – SECONDA PROVA SCRITTA ANNO 2004)

(2) 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:

4. implementi in linguaggio SQL le seguenti interrogazioni:

Q1: Data una classe ed un anno scolastico, visualizzare quali studenti di quella classe hanno frequentato corsi e di che tipo;

Q2: Dato uno studente, visualizzare quali corsi ha frequentato, di che tipo, per quale monte ore e in quale anno scolastico;

Q3: Dato un anno scolastico, visualizzare quali corsi interni sono stati attivati e da quali studenti sono stati seguiti;

Q4: Dato un corso, visualizzare quali sono i dati relativi ad esso e per quali anni scolastici è stato attivato;

Q5: Dato uno studente, visualizzare quali classi ha frequentato, in quali anni scolastici e con quali esiti finali;

Q6: Per ogni anno scolastico, contare il numero di studenti respinti;

Q7: Dato un anno scolastico, contare il numero totale di ore dei corsi organizzati per l'arricchimento dell'offerta formativa;

Q8: Visualizzare l'elenco degli studenti che non hanno mai frequentato corsi di recupero.

Svolgimento

Premessa: omessa (vedi svolgimento esercizio precedente)

Da un'attenta lettura delle specifiche si evidenzia che sono richieste le seguenti attività:

- gestione degli studenti e della loro situazione scolastica;
- gestione dei corsi interni ed esterni alla scuola.

Possiamo quindi ipotizzare di avere entità contenenti le informazioni relative:

- agli *studenti*
- alle *classi* da loro frequentate con le risultanze finali anno per anno;
- ai *corsi* da loro frequentati (suddivisi in *interni* ed *esterni*)

Abbiamo quindi ottenuto un primo elenco di entità che dovranno entrare a far parte dello schema concettuale della base di dati che si sta progettando.

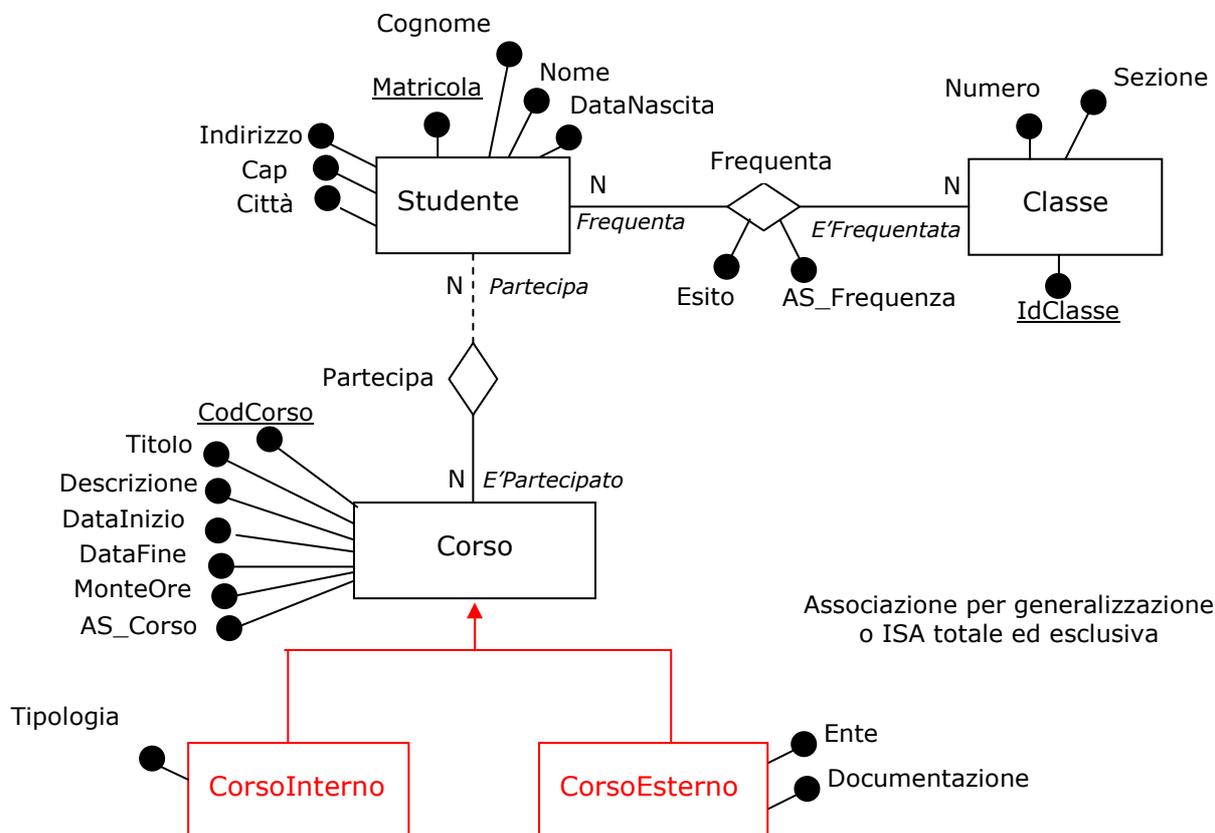
A partire da questo elenco, associamo a ciascuna entità individuata i corrispondenti attributi, attingendo sempre le informazioni dalle specifiche fornite:

- a ciascuno studente devono essere associati una *Matricola*, un *Cognome*, un *Nome*, una *DataNascita*, un *Indirizzo*, un *Cap*, una *Città* (anagrafica breve);
- a ciascuna classe deve essere associata la coppia di valori *numero ordinale classe* e *lettera sezione*;
- ciascun corso è caratterizzato da un *Titolo*, una *Descrizione*, una *DataInizio*, una *DataFine*, un *MonteOre* e dal *AnnoScolastico* di riferimento in cui stato attivato; ai corsi interni è inoltre associata una *Tipologia* (corso di recupero, di sostegno, di arricchimento) mentre a quelli esterni è associato l'*Ente* organizzatore e la *Documentazione* di accertamento.

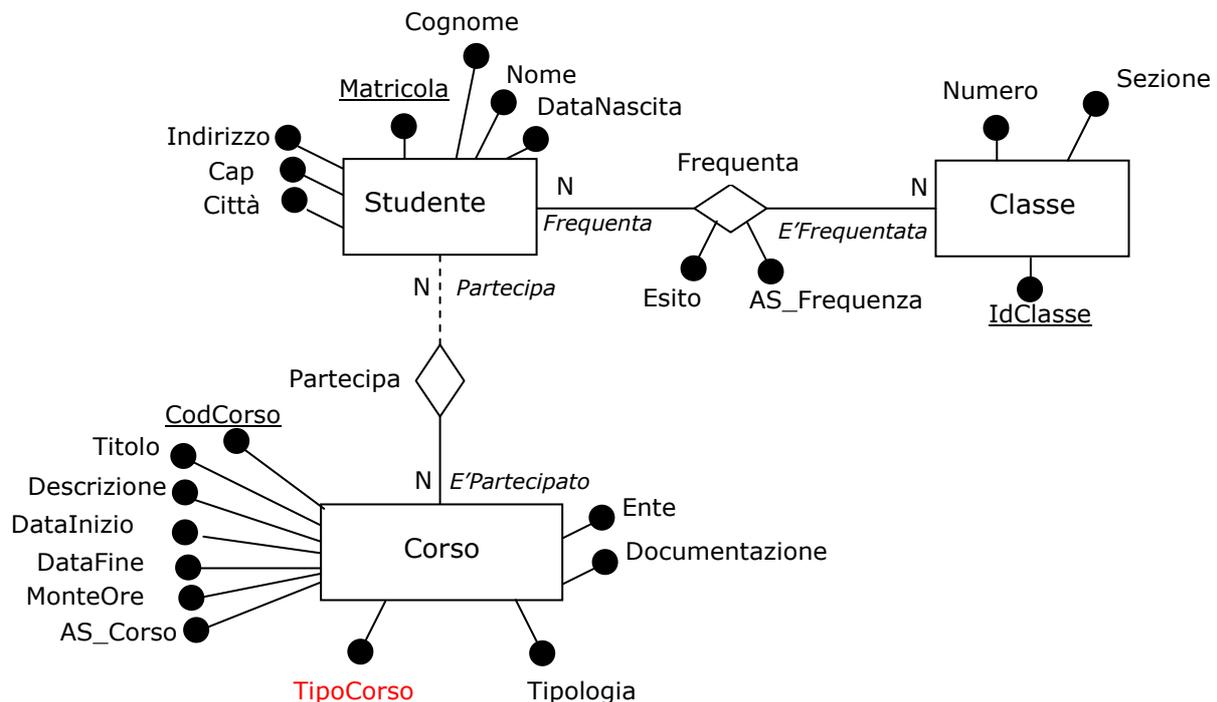
Passiamo ora ad esaminare quali sono le associazioni tra le entità ipotizzate, individuando per ciascuna di esse la molteplicità dell'associazione in base alle caratteristiche di funzionalità evidenziate nelle specifiche più eventuali attributi:

- esiste un'associazione ISA totale ed esclusiva con la funzione di suddividere le istanze dell'entità padre *Corso* nelle entità figlie *CorsoInterno* e *CorsoEsterno*, in modo da poter specificare per ciascuna sottoentità i corrispondenti attributi aggiuntivi;
- tra le entità *Studente* e *Corso* esiste un'associazione "Partecipa" di molteplicità N:N (con diretta parziale ed inversa totale) in quanto "uno studente può partecipare a nessuno o più corsi e viceversa un corso deve essere partecipato da uno o più studenti";
- tra le entità *Studente* e *Classe* esiste un'associazione "Frequenta" di molteplicità N:N (totale in entrambi i versi) in quanto "uno studente deve frequentare una o più classi (si tiene conto cos' dello "storico" ossia delle classi da lui frequentate negli anni precedenti) ed ogni classe deve essere frequentata da uno o più studenti"; sono inoltre mantenute come attributi dell'associazione, le informazioni sull'Esito (supponendo che possa assumere il valore "Promosso", "Respinto", "Debito") conclusivo di ogni anno scolastico per ciascuno studente nella corrispondente classe frequentata; se si desidera si potrebbe mantenere l'elenco dei debiti formativi (attributo multiplo opzionale) introducendo un'altra entità *Materia* ed un'altra associazione

Schema concettuale della realtà di interesse – diagramma E/R con ISA



**(a) Schema concettuale della realtà di interesse – diagramma E/R senza ISA
(ottenuto attraverso l'accorpamento delle entità figlie nell'entità padre perché l'ISA è TOTALE)**



Vincoli impliciti:

Sono quei vincoli direttamente deducibili dal diagramma E/R in quanto caratterizzati graficamente:

- vincoli di chiave primaria: tutti gli attributi sottolineati
- vincoli di integrità referenziale: totalità delle associazioni dirette e/o inverse (linea continua)

Vincoli espliciti:

Vincoli non deducibili direttamente dal diagramma E/R:

V1: (Corso.Tipologia="Recupero") OR (Corso.Tipologia="Sostegno") OR (Corso.Tipologia="Arricchimento")

V2: (Corso.Tipologia="Interno") OR (Corso.Tipologia="Esterno")

V3: (Classe.Numero=1) OR (Classe.Numero=2) OR (Classe.Numero=3) OR (Classe.Numero=4) OR (Classe.Numero = 5)

V4: (Classe.Sezione='A') OR (Classe.Sezione='B') OR (Classe.Sezione='C') OR (Classe.Sezione='D') OR (Classe.Sezione='E') OR (Classe.Sezione='F') OR (Classe.Sezione='G')

V5: (Frequenta.Esito="Promosso") OR (Frequenta.Esito="Respinto") OR (Frequenta.Esito="Debito")

V6: (Corso.DataInizio < Corso.DataFine)

V7: (Corso.MonteOre > 0)

V8: (Corso.DataInizio > Studente.DataNascita)

V9: SE (Corso.TipoCorso = "Interno")

ALLORA

Corso.Ente = NULL

Corso.Documentazione = NULL

FINE SE

V10: SE (Corso.TipoCorso = "Esterno")

ALLORA

Corso.Tipologia = NULL

FINE SE

(b) Schema logico della realtà di interesse - mapping relazionale del diagramma E/R

(i) mapping dell'associazione "SiAbbona" di molteplicità N:N tra le entità "Abbonato" e "Pubblicazione"

Studente (Matricola, Cognome, Nome, DataNascita, Indirizzo, Cap, Citta)

Classe (IdClasse, Numero, Sezione)

Frequenta (Matricola1, IdClasse1, Esito, AS_Frequenza)

con "Matricola1" chiave esterna (foreign key) sull'attributo "Matricola" della relazione "Studente"

con "IdClasse1" chiave esterna (foreign key) sull'attributo "IdClasse" della relazione "Classe"

$VR_{Matricola1} (Frequenta) \subseteq VR_{Matricola} (Studente)$

dal mapping relazionale dell'associazione N:N

$VR_{IdClasse1} (Frequenta) \subseteq VR_{IdClasse} (Classe)$

dal mapping relazionale dell'associazione N:N

$VR_{Matricola} (Studente) \subseteq VR_{Matricola1} (Frequenta)$

dalla TOTALITA' dell'associazione diretta "Frequenta"

$VR_{IdClasse} (Classe) \subseteq VR_{IdClasse1} (Frequenta)$

dalla TOTALITA' dell'associazione inversa "E'Frequentata"

(ii) mapping dell'associazione "Partecipa" di molteplicità N:N tra le entità "Studente" e "Corso"

Studente già mappato in precedenza

Corso (CodCorso, Titolo, Descrizione, DataInizio, DatFine, MonteOre, AS_Corso, Tipologia, Ente, Documentazione, TipoCorso)

Partecipa (Matricola2, CodCorso2)

con "Matricola2" chiave esterna (foreign key) sull'attributo "Matricola" della relazione "Studente"

con "CodCorso2" chiave esterna (foreign key) sull'attributo "CodCorso" della relazione "Corso"

$VR_{Matricola2} (Partecipa) \subseteq VR_{Matricola} (Studente)$

dal mapping relazionale dell'associazione N:N

$VR_{CodCorso2} (Partecipa) \subseteq VR_{CodCorso} (Corso)$

dal mapping relazionale dell'associazione N:N

$VR_{CodCorso} (Corso) \subseteq VR_{CodCorso2} (Partecipa)$

dalla TOTALITA' dell'associazione inversa "E'Partecipato"

Vincoli di integrità intrarelazionali (o interni) ed interrelazionali (o esterni)

I vincoli di chiave primaria (impliciti nel modello E/R) sono mappati in *vincoli intrarelazionali su più ennuple*.

I vincoli di totalità di un'associazione (impliciti nel modello E/R) sono mappati in *vincoli interrelazionali di tipo referenziale*

I vincoli espliciti del diagramma E/R vengono mappati come segue:

V1 (Corso): "(Tipologia="Recupero") OR (Tipologia="Sostegno") OR (Tipologia="Arricchimento")" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V2 (Corso): "(Tipologia="Interno") OR (Tipologia="Esterno")" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V3 (Classe): "(Numero=1) OR (Numero=2) OR (Numero=3) OR (Numero=4) OR (Numero = 5)" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V4 (Classe): "(Sezione='A') OR (Sezione='B') OR (Sezione='C') OR (Sezione='D') OR (Sezione='E') OR (Sezione='F') OR (Sezione='G')" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V5 (Frequenta): "(Esito="Promosso") OR (Esito="Respinto") OR (Esito="Debito")" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V6 (Corso): "(DataInizio < DataFine)" *vincolo intrarelazionale su singola ennupla su più di un attributo*

V7 (Corso): "(MonteOre > 0)" *vincolo intrarelazionale su singola ennupla su singolo attributo*

V8 (Corso, Studente): "(Corso.DataInizio > Studente.DataNascita)" *vincolo interrelazionale di tipo non referenziale*

V9 (Corso): SE (TipoCorso = "Interno") *vincolo intrarelazionale su singola ennupla su più di un attributo*

ALLORA

Ente = NULL

Documentazione = NULL

FINE SE

V10 (Corso): SE (TipoCorso = "Esterno") *vincolo intrarelazionale su singola ennupla su più di un attributo*

```

    ALLORA
        Tipologia = NULL
    FINE SE

```

Normalizzazione: omessa (vedi svolgimento esercizio precedente)

(c) Definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL

CREATE DATABASE Scuola;

USE Scuola;

CREATE DOMAIN MiaTipologia **AS** CHAR(13)

CHECK (VALUE IN ("Recupero", "Sostegno", "Arricchimento")); // Vincolo V1

CREATE DOMAIN MioTipo **AS** CHAR(7)

CHECK (VALUE IN ("Interno", "Esterno")); // Vincolo V2

CREATE DOMAIN MioEsito **AS** CHAR(8)

CHECK (VALUE IN ("Promosso", "Respinto", "Debito")); // Vincolo V5

CREATE TABLE Studente

```

(
Matricola          CHAR(10)          NOT NULL,
Cognome            CHAR(50)          NOT NULL,
Nome               CHAR(50)          NOT NULL,
DataNascita        DATE              NOT NULL,
Indirizzo          CHAR(255)         NOT NULL,
Cap                CHAR(5)           NOT NULL,
Citta              CHAR(30)          NOT NULL,
PRIMARY KEY (Matricola)
);

```

CREATE TABLE Classe

```

(
IdClasse           INT              NOT NULL,
Numero             SMALLINT         NOT NULL,
Sezione           CHAR(1)           NOT NULL,
PRIMARY KEY (IdClasse),
CHECK (Numero VALUE IN (1, 2, 3, 4, 5)), //Vincolo V3
CHECK (Sezione VALUE IN ('A', 'B', 'C', 'D', 'E', 'F', 'G')) // Vincolo V4
);

```

CREATE TABLE Frequenta

```

(
Matricola1         CHAR(10)         NOT NULL,
IdClasse1          INT              NOT NULL,
Esito              MioEsito         NOT NULL,
AS_Frequenta       CHAR(9)          NOT NULL,
PRIMARY KEY (Matricola1, IdClasse1),
FOREIGN KEY (Matricola1) REFERENCES ON Studente (Matricola) // VR di chiave esterna
[ON DELETE NO ACTION,] ON UPDATE CASCADE,
FOREIGN KEY (IdClasse1) REFERENCES ON Classe (IdClasse) // VR di chiave esterna
[ON DELETE NO ACTION,] ON UPDATE CASCADE );

```

CREATE TABLE Corso

```
(
CodCorso          CHAR(10)          NOT NULL,
Titolo            CHAR(50)          NOT NULL,
Descrizione       CHAR(255)        NOT NULL,
DataInizio        DATE              NOT NULL,
DataFine          DATE              NOT NULL,
MonteOre          SMALLINT         NOT NULL,
AS_Corso          CHAR(9)           NOT NULL,
Tipologia         MiaTipologia      NOT NULL,
Esito             MioEsito          NOT NULL,
Descrizione       VARCHAR(1000)     NOT NULL,
PRIMARY KEY (CodCorso),
CHECK (DataInizio < DataFine),           //Vincolo V6
CHECK (MonteOre > 0)                     //Vincolo V7
);
```

CREATE TABLE Partecipa

```
(
Matricola2        CHAR(10)          NOT NULL,
CodCorso2         CHAR(10)          NOT NULL,
PRIMARY KEY (Matricola2, CodCorso2),
FOREIGN KEY (Matricola2) REFERENCES ON Studente (Matricola) // VR di chiave esterna
[ON DELETE NO ACTION,] ON UPDATE CASCADE,
FOREIGN KEY (CodCorso2) REFERENCES ON Corso (CodCorso) // VR di chiave esterna
[ON DELETE NO ACTION,] ON UPDATE CASCADE
);
```

CREATE ASSERTION V8 CHECK (Corso.DataInizio > Studente.DataNascita);

(d) Svolgimento delle query prima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL

Q1. Data una classe ed un anno scolastico, visualizzare quali studenti di quella classe hanno frequentato corsi e di che tipo

SQL dapprima occorre connettersi al database e poi

USE Scuola;

```
SELECT Cognome, Nome, Titolo, Descrizione, DataInizio, DataFine
FROM Studente, Frequenta, Classe, Partecipa, Corso
WHERE (Studente.Matricola = Frequenta.Matricola1) AND (Frequenta.IdClasse1 = Classe.IdClasse)
AND (Studente.Matricola=Partecipa.Matricola2) AND (Partecipa.CodCorso2 = Corso.CodCorso)
AND (Classe.Numero =[InserisciNumero]) AND (Classe.Sezione =[InserisciSezione])
AND (AS_Frequenta = [InserisciAS]);
```

N.B. e' possibile ordinarli per Cognome e Nome

Q2. Dato uno studente, visualizzare quali corsi ha frequentato, di che tipo, per quale monte ore e in quale anno scolastico

SQL dapprima occorre connettersi al database e poi

USE Scuola;

SELECT Titolo, Descrizione, DataInizio, DataFine, MonteOre, AS_Corso

FROM Partecipa, Corso

WHERE (Partecipa.CodCorso2 = Corso.CodCorso)

AND (Partecipa.Matricola2 = [InserisciMatricola]);

oppure

SELECT Titolo, Descrizione, DataInizio, DataFine, MonteOre, AS_Corso

FROM Studente, Partecipa, Corso

WHERE (Studente.Matricola = Partecipa.Matricola2) **AND** (Partecipa.CodCorso2 = Corso.CodCorso)

AND (Studente.Cognome = [InserisciCognome]) **AND** (Studente.Nome = [InserisciNome]);

Q3. Dato un anno scolastico, visualizzare quali corsi interni sono stati attivati e da quali studenti sono stati seguiti

SQL dapprima occorre connettersi al database e poi

USE Scuola;

N.B. e' possibile ordinarli per CodCorso

SELECT CodCorso, Titolo, Tipologia, Cognome, Nome

FROM Studente, Partecipa, Corso

WHERE (Studente.Matricola=Partecipa.Matricola2) **AND** (Partecipa.CodCorso2 = Corso.CodCorso)

AND (AS_Corso = [InserisciAS]) **AND** (Tipologia = "Interno");

Q4. Dato un corso, visualizzare i dati relativi ad esso e per quali anni scolastici è stato attivato

SQL

SELECT CodCorso, Descrizione, Data Inizio, DataFine, MonteOre, AS_Corso

FROM Corso

WHERE (Titolo = [InserisciTitolo]);

Q5. Dato uno studente, visualizzare quali classi ha frequentato, in quali anni scolastici e con quali esiti finali

SQL dapprima occorre connettersi al database e poi

USE Scuola;

N.B. e' possibile ordinarli per anno scolastico (AS_frequenta)

SELECT AS_Frequenta, Numero, Sezione, Esito

FROM Frequenta, Classe

WHERE (Frequenta.IdClasse1 = Classe.IdClasse)

AND (Frequenta.Matricola1 = [Inserisci Matricola]);

oppure

SELECT AS_Frequenta, Numero, Sezione, Esito

FROM Studente, Frequenta, Classe

WHERE (Studente.Matricola = Frequenta.Matricola1) **AND** (Frequenta.IdClasse1 = Classe.IdClasse)

AND (Studente.Cognome = [InserisciCognome]) **AND** (Studente.Nome = [InserisciNome]);

Q6. Per ogni anno scolastico, contare il numero di studenti respinti

SQL dapprima occorre connettersi al database e poi

USE Scuola;

SELECT AS_Frequenta, COUNT(*) As NumRespinti

FROM Frequentata

WHERE (Esito = "Respinto")

GROUP BY AS_Frequenta;

N.B. e' possibile ordinarli per anno scolastico (AS_Frequenta)

Q7. Dato un anno scolastico, contare il numero totale di ore dei corsi organizzati per l'arricchimento dell'offerta formativa

SQL dapprima occorre connettersi al database e poi

USE Scuola;

SELECT SUM(MonteOre) AS TotOre

FROM Corso

WHERE (Tipologia = "Arricchimento")

AND (AS_Corso = [InserisciAS]);

Q8. Visualizzare l'elenco degli studenti che non hanno mai frequentato corsi di recupero

SQL dapprima occorre connettersi al database e poi

USE Scuola;

(**SELECT** Matricola, Cognome, Nome

FROM Studente)

MINUS

(**SELECT DISTINCT** Matricola, Cognome, Nome

FROM Studente, Partecipa, Corso

WHERE (Studente.Matricola=Partecipa.Matricola2) **AND** (Partecipa.CodCorso2 = Corso.CodCorso)

AND (Tipologia = "Recupero");

(3) Una biblioteca vuole realizzare una base dati per gestire le sue attività di classificazione, ricerca e prestito dei libri ai suoi soci.

Per ogni socio si vogliono registrare i dati anagrafici e per ogni libro si vuole archiviare il titolo, l'autore, l'editore e l'anno di pubblicazione. Inoltre si vogliono registrare le informazioni relative alla collocazione del libro nella biblioteca, ai suoi contenuti (attraverso parole chiave) ed ai prestiti del libro ai soci.

Per ogni libro esistono più copie in biblioteca ed un socio può prendere in prestito anche più di un libro contemporaneamente. È fissato a 15 il numero dei giorni del prestito.

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Dato il nominativo di un autore, visualizzare i libri da lui scritti presenti in biblioteca;
Q2: Dato il nominativo di un socio, visualizzare i libri attualmente in prestito;
Q3: Data una parola chiave, ricercare i libri che la contengono;
Q4: Per ogni autore, determinare il numero dei libri presenti in biblioteca;
Q5: Dato il titolo di un libro, determinare il numero di copie presenti in biblioteca;
Q6: Determinare il numero di copie di libri attualmente in prestito;
Q7: Visualizzare il codice dei soci che attualmente non hanno libri in prestito;
Q8: Visualizzare l'elenco dei libri di cui esistono più copie in biblioteca.

(4) In un istituto scolastico si vogliono organizzare degli scambi tra un gruppo di studenti dell'istituto ed un gruppo di studenti stranieri al fine di migliorare la conoscenza delle lingue.

Per realizzare gli scambi ed al fine di scegliere un'opportuna sistemazione presso una famiglia straniera, si raccolgono presso gli studenti interessati i seguenti dati: cognome, nome, data di nascita, classe e sezione frequentata, numero di fratelli e sorelle, tipo di professione esercitata dal padre e dalla madre.

Ogni scambio è individuato da un codice, la scuola straniera collegata, la nazione di appartenenza, il numero di studenti coinvolti, la data di inizio e di fine, l'anno scolastico in cui avviene e l'elenco degli studenti che vi partecipano.

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Elenco degli studenti che hanno effettuato un determinato scambio con una data scuola;
Q2: Elenco di tutti gli scambi effettuati nello stesso anno scolastico;
Q3: Elenco degli studenti il cui padre esercita una determinata professione;
Q4: Cognome e nome di tutti gli studenti che hanno partecipato ad uno scambio con scuole del Portogallo.

(5) Le informazioni relative alle **attività sportive studentesche** devono essere organizzate in una base dati. Gli **studenti**, dei quali si conservano le informazioni anagrafiche, frequentano gli **istituti superiori**, e possono partecipare ad una o più **manifestazioni sportive** (specialità sportive diverse, giornate diverse, campionati che durano mesi o gare di un giorno).

Per ogni attività sportiva le scuole indicano un **professore** che svolge la funzione di riferimento e di allenatore: ogni professore segue una sola manifestazione, ma una stessa manifestazione può essere seguita da professori diversi di scuole diverse.

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Numero degli studenti che partecipano ad una determinata manifestazione sportiva;
Q2: Elenco anagrafico degli allenatori di un'attività sportiva;
Q3: Elenco delle scuole (denominazione) con il numero di studenti che partecipano alle attività sportive;
Q4: Elenco delle scuole (con denominazione, indirizzo, telefono) con studenti che partecipano ad una determinata manifestazione sportiva;
Q5: Elenco allenatori (cognome e nome) e scuole (denominazione) di appartenenza in ordine alfabetico;
Q6: Numero degli studenti partecipanti ad una determinata scuola per ciascuna delle manifestazioni sportive.

(6) Una **galleria d'arte** ha deciso di creare un sistema che **consenta via web ai suoi clienti registrati** di consultare **il catalogo completo dei quadri** in listino ed ad un proprio utente amministratore di inserire, modificare o cancellare le informazioni relative agli stessi.

Per ogni quadro presente in galleria è compilata una scheda che riporta l'autore, il titolo, la tipologia tecnica realizzativa (olio, tempera, carboncino, litografia, etc.), le dimensioni, il prezzo, nonché l'immagine illustrativa dell'opera.

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Elencare tutte le opere realizzate con una determinata tecnica pittorica;
Q2: Elencare tutte le opere realizzate da un determinato artista il cui prezzo è inferiore ai 300 euro;

Inoltre si provi a progettare ed a sviluppare in dettaglio utilizzando un linguaggio di programmazione lato server una delle seguenti funzioni:

- (-) pagina di accesso all'area riservata ai clienti registrati;
- (-) pagina riservata ai clienti per la consultazione del catalogo e delle singole schede delle opere;
- (-) pagina riservata all'amministratore per la gestione completa dei dati presenti in catalogo.

(7) Il nuovo **direttore finanziario** di una **società** intende automatizzare **la gestione dei pagamenti dei crediti erogati ai propri clienti**. Ogni credito è identificato con codice, denominazione del cliente, indirizzo, provincia e regione di appartenenza, modalità di pagamento (pronta cassa, 30 giorni, 60 giorni o 90 giorni), ammontare del credito stesso, data di concessione del credito, data del pagamento prestabilita, esito del pagamento (non effettuato perché ancora non si è raggiunti la data di pagamento prevista, andato a buon fine perché pagato entro i limiti, non andato a buon fine se il cliente non ha pagato entro la data prestabilita).

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Elenco dei clienti che devono ancora effettuare i pagamenti per ogni provincia e per ogni regione;
Q2: Elenco dei clienti che devono effettuare il pagamento in un certo giorno;
Q3: Per ogni tipo di pagamento, elenco dei clienti che hanno scelto quel tipo di pagamento;
Q4: Elenco dei clienti che hanno ricevuto un credito superiore a 100.000 euro e non hanno effettuato il pagamento entro la data stabilita (cattivi pagatori).

(8) Un'**azienda** vuole gestire **gli ordini dei propri clienti relativi ai prodotti** trattati in modo automatizzato. Ogni ordine è individuato da codice, data, valore complessivo in euro, denominazione cliente, indirizzo, comune e provincia ed è costituito dall'intenzione di acquisto in quantità variabili di determinati prodotti.

Ogni cliente è individuato dalla sua anagrafica.

Ogni prodotto è individuato da codice, descrizione, prezzo di listino

Si realizzino, fatte le ipotesi aggiuntive del caso,

- a) Uno schema concettuale della realtà di interesse attraverso la produzione del diagramma E/R (scrivendo esplicitamente le conseguenti regole di lettura);
- b) lo schema logico della realtà di interesse ottenuto attraverso il mapping relazionale dello schema concettuale (diagramma E/R) ottenuto al punto precedente;
- c) la definizione delle relazioni della base dati ottenute al punto precedente in linguaggio SQL.

Ed inoltre

- d) si implementino, dapprima utilizzando gli operatori dell'algebra relazionale (se possibile) poi usando il linguaggio SQL, le seguenti interrogazioni:
Q1: Visualizzare il numero di ordini effettuati da ogni cliente;
Q2: Visualizzare il numero di ordini effettuati complessivamente da tutti i clienti;
Q3: Visualizzare il valore medio degli ordini di ciascun cliente;
Q4: Elenco dei clienti che hanno un valore complessivo degli ordini al di sotto della media.