

Consiglio Nazionale delle Ricerche

Il sistema informativo INFEA

Progettazione della base di dati

Nicola Aloia

Rapporto
CNUCE-B4-2002-005

CNUCE

Pisa

Il sistema informativo INFEA

Progettazione della base di dati

Nicola Aloia

CNUCE - Istituto del C.N.R.

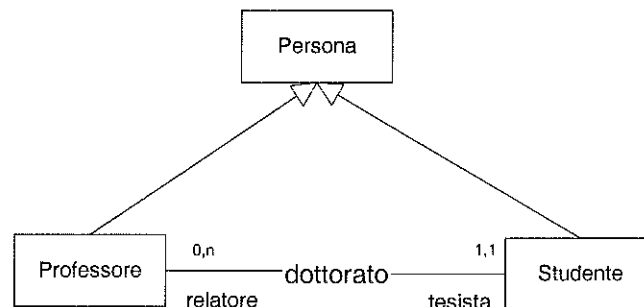
Introduzione

Il presente rapporto descrive la base di dati di supporto alla componente informatica del sistema informativo INFEA, progettato e realizzato per il Servizio Sviluppo Sostenibile del Ministero dell'Ambiente. Per informazioni e dettagli sulle caratteristiche e le funzionalità del sistema informativo si rimanda alla bibliografia citata a fine documento. Ciò che qui vogliamo evidenziare sono i dettagli relativi alla fase di progettazione concettuale e progettazione logico fisica della base di dati, al fine di fornire una documentazione di riferimento per ulteriori sviluppi e approfondimenti sia per la prosecuzione dell'attività finanziata dal Ministero, mirante all'evoluzione del sistema *Informativo per la Formazione e l'Educazione Ambientale* (INFEA) nel *Sistema Informativo per il Servizio Sviluppo Sostenibile*, che contempra strumenti di governo interno, sia per la gestione di alcune pratiche che per le nuove esigenze di distribuzione di informazione correlate alla legge sulla trasparenza, sia per quanti a livello regionale o provinciale di trovassero ad affrontare problematiche connesse, o simili, a quelle trattate in Infea.

Nel capitolo 1 è illustrata la fase di progettazione concettuale della base di dati e nel capitolo 2 la fase di progettazione logico fisica. L'analisi dei requisiti dei dati e delle procedure si è basata sulla documentazione disponibile per i sistemi "ANFORA", "ANDREA" e delle basi di dati "il Nostro mondo" e "la settimana dell'educazione ambientale", che hanno costituito l'ossatura iniziale della base di dati per INFEA, e sulle continue interazioni con i gruppi del Ministero dell'Ambiente e dell'ISFOL. In questo documento non viene fornita una descrizione dell'analisi dei requisiti, che ha determinato lo schema concettuale risultante, ma per ogni classe dello schema vengono descritte le motivazioni e le scelte adottate nella modellazione.

1. Progettazione concettuale della base di dati

La costruzione dello schema concettuale consiste in un processo di conoscenza della realtà di interesse, e in una traduzione della conoscenza acquisita in termini di un insieme prefissato di strutture di rappresentazione. Nei diagrammi di Fig. 1 e Fig. 2 è mostrato lo schema concettuale relativo alla base di dati per il sistema informativo INFEA. Il formalismo grafico adottato è un sottoinsieme dell'UML (Unified Modeling Language - strumento diffuso nell'attività di progettazione concettuale), in cui i rettangoli rappresentano le *classi* o *entità* (cioè insiemi di oggetti del mondo reale per cui s'intende registrare dati per ricavarne informazione), gli archi che uniscono due classi rappresentano le *associazioni* (cioè leggi di corrispondenza tra gli insiemi), al centro dell'arco può esserci un nome che chiarisce il significato dell'associazione, ci possono inoltre essere dei nomi agli estremi dell'arco per chiarire il *ruolo* che una certa classe ha nell'associazione; le coppie di numeri agli estremi degli archi rappresentano il numero minimo e massimo di elementi della classe che partecipano nell'associazione; gli archi che terminano con una freccia rappresentano *gerarchie di specializzazione*, o gerarchia IS-A, (cioè la classe più specializzata possiede tutte le proprietà della sua super classe più altre proprie). Ad esempio, il seguente diagramma



si legge nel seguente modo: le classi *Professore* e *Studente* sono entrambe (IS-A) sottoclassi della classe *Persona*, cioè possiedono tutte le proprietà di *Persona* (es. Nome, Cognome, Indirizzo, etc) più alcune specifiche (es. la proprietà: *dataNomina* per la classe *Professore* e la proprietà *NumeroMatricola* per la classe *Studente*). Tra le classi *Professore* e *Studente* esiste un'associazione di nome *dottorato* (cioè una legge, che fa corrispondere ad elementi della classe *Professore*, elementi della classe *Studente* e viceversa). Gli elementi della classe *Studente* partecipano minimo 1 e massimo 1 volte all'associazione *dottorato* (cioè si è supposto che facciano riferimento obbligatoriamente ad un solo professore), mentre gli elementi della classe *Professore* partecipano all'associazione minimo 0 volte (non è obbligatorio avere dei dottorandi) e massimo *n* (coloro che seguono contemporaneamente più dottorandi). Nell'associazione *dottorato* i professori giocano il ruolo di *relatore* e gli studenti giocano il ruolo di *tesista*.

Per evitare l'incrociarsi di archi e rendere più leggibile il disegno, sono stati introdotti, come estensione del formalismo, cerchi di riferimento, a bordo doppio, e cerchi di aggancio, a bordo singolo, che contengono all'interno una lettera. Il cerchio di riferimento è il punto di arrivo di ipotetici archi partenti da tutti i cerchi di aggancio

che contengono la stessa lettera (es. in Fig.1 il cerchio contenente la lettera "L" posto sotto la classe *Locazione* è il riferimento dei cerchi di aggancio contenenti la lettera "L" posti accanto alle classi *Servizio, Manifestazione, Visita e Soggiorno*, ecc. infatti tutte queste classi hanno un'associazione con la classe *Locazione*).

Per gli stessi obiettivi di leggibilità sono stati introdotti rettangoli a bordo doppio per indicare che una certa classe viene descritta più dettagliatamente in altro disegno (es. la classe *Offerta Formativa* di Fig.1 è descritta nei dettagli in Fig.2).

10

11

12

13

14

15

(*) L'associazione ha come proprietà la "Data di adesione alla carta di Aalborg"
 (**) L'associazione ha come proprietà "Quota Finanziata" e "Percentuale rispetto al costo totale"

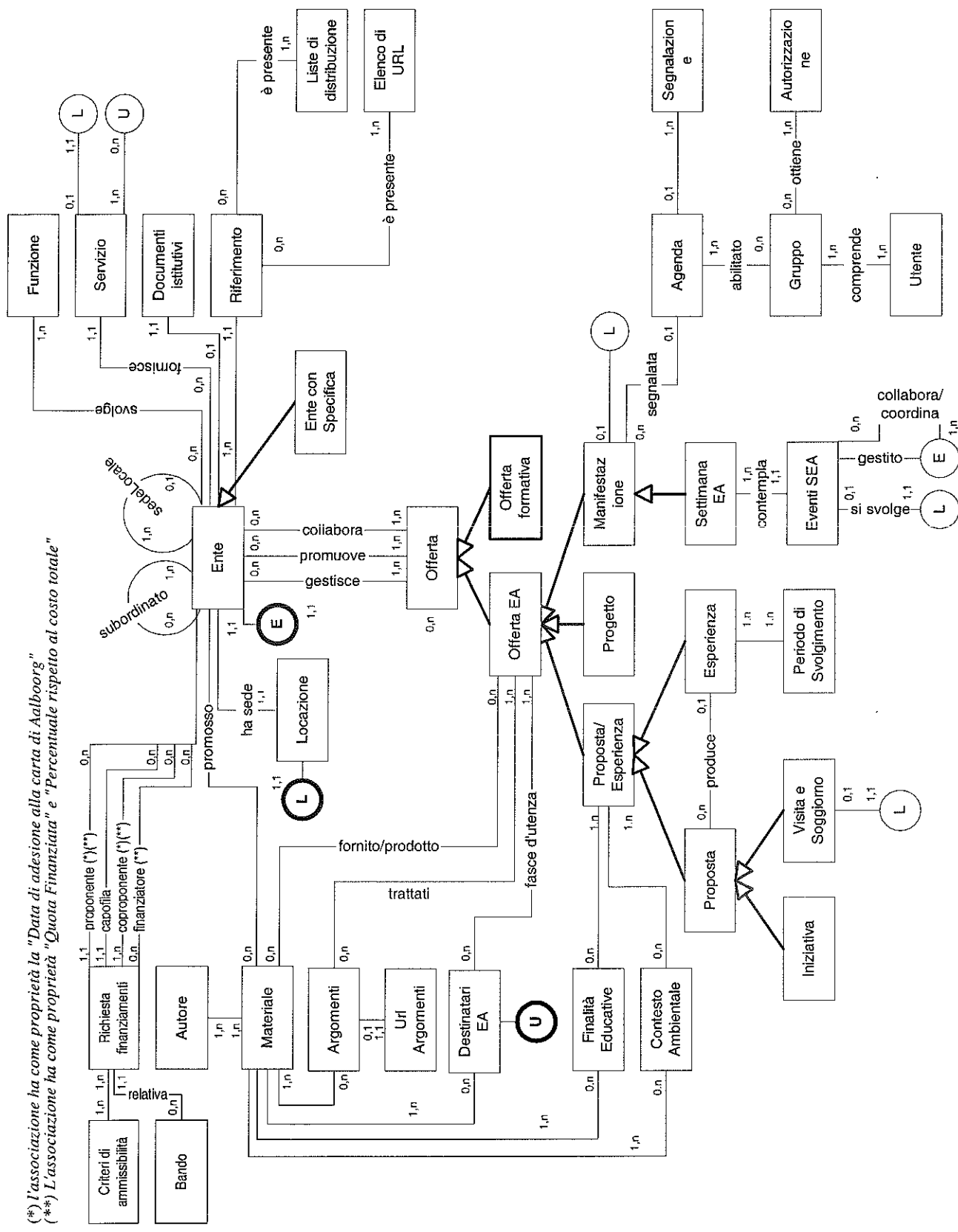


Figure 1 Schema Concettuale

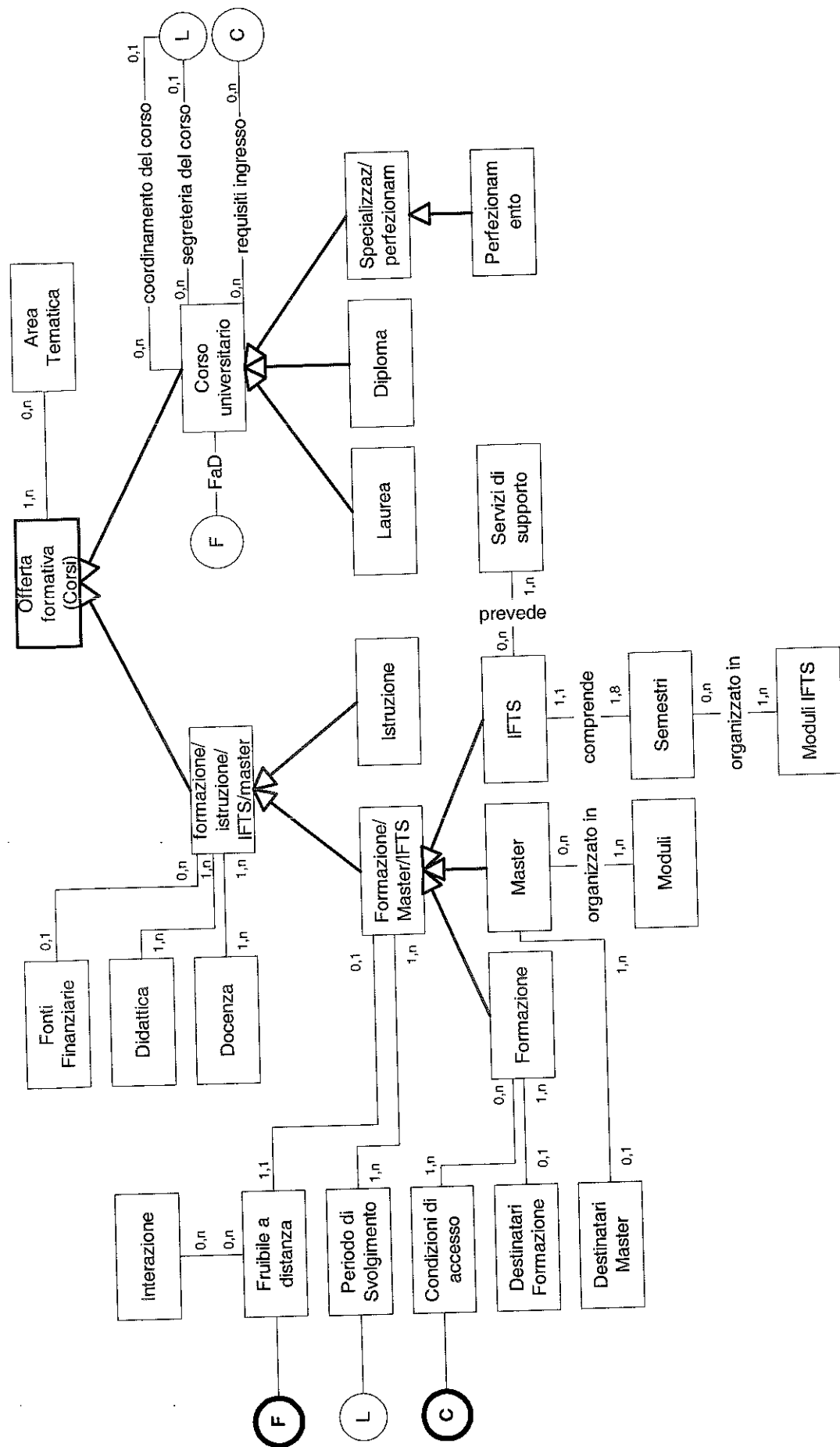


Fig.2 Schema Concettuale: dettaglio della classe Offerta Formativa

Descrizione dettagliata delle Classi (Entità):

In questo paragrafo sono descritti nei dettagli i diagrammi di Fig. 1 e Fig.2. Per ogni classe dopo una breve descrizione è presentato l'elenco delle proprietà che la caratterizzano. Per ogni proprietà viene dato un nome (in neretto), un tipo (in corsivo) ed una breve descrizione o commento.

Classe Ente: questa categoria concettuale è presente in tutte le realtà da noi osservate (Andrea, Anfora, il Nostro mondo, la settimana dell'educazione ambientale), con livello di dettagli (proprietà) più o meno approfonditi. La soluzione proposta è un'astrazione che contempla tutte le situazioni.

Elenco proprietà:

1. **IdEnte** –*codice*- E' un identificatore interno univoco per ogni ente. Occorre implementare un meccanismo per l'assegnazione automatica del valore. Questa proprietà non è mai visibile all'utilizzatore.
 2. **Nome** –*stringa* –
 3. **TipologiaIstituzionale** – *codifica* – Macroclassificazione degli enti, può assumere i seguenti valori.
 - **EntiPubbliciCentrali** – vedi classe *EnteConSpecifica*.
 - **EntiPubbliciterritoriali** – vedi classe *EnteConSpecifica*
 - **EnteUniversitario** – vedi classe *EnteUniversitario*
 - **ServizioCulturale** Questa tipologia raggruppa vari enti, la cui specifica è direttamente richiamata dal valore della proprietà *Nome*. Esempi: "Fondazione G.Agnelli", oppure "Biblioteca Labronica" in quest'ultimo caso però l'ente può essere presente solo se organizza qualcosa attinente all'ambiente (es. una mostra) non in quanto fornitore di servizi (servizio di biblioteca) per l'ente "Comune di Livorno". Vedi classe *EnteConSpecifica*.
 - **Scuola** – vedi classe *EnteConSpecifica*.
 - **EntiEdIstitutiDiRicercaPubbliciEPrivati** – Istituti ed enti di ricerca pubblici (CNR, Enea, Isfol, Istat,etc) o privati (es Istituto per l'ambiente, CENSIS, Consorzi).
 - **Agenzia/Authority** – vedi classe *EnteConSpecifica*.
 - **AreaServiziDiPubblicaUtilità** – Enti pubblici o privati con funzioni di gestione, controllo su tematiche di pubblico interesse. Vedi classe *EnteConSpecifica*
 - **PartiSocialiEdEntiDiLoroEmanazione** – Sindacati, associazioni di categoria, ecc..
 - **AreaProtetta** - Ente istituito con decreto statale a fini di protezione ambientale. Vedi classe *EnteConSpecifica*.
 - **TerzoSettore** – Organizzazioni senza fini di lucro. Vedi classe *EnteConSpecifica*
 - **OrganismiInternazionali** – (FAO, CE, UNESCO,...)
 - **Impresa** (SoftwareHouse, CaseEditrici,)
 4. **Sito Web** – *stringa* -
 5. **Descrizione** – *testo* – contiene in formato libero una descrizione dell'Ente. Non si prevede di fare ricerche in base al contenuto del testo.
 6. **Logo** – *immagine* -
-

Classe Riferimenti: La ragione di questa classe è di consentire la realizzazione di elenchi di persone/istituzioni/etc correlate a uno stesso ente e per le quali non si ritiene opportuno l'inserimento di una nuova istanza di Ente eventualmente subordinato. L'utilizzo primario è la gestione di un indirizzario dell'amministrazione INFEA. La classe riferimento ha relazione (1,n) con la classe Ente. Per ogni ente è comunque dato almeno un riferimento e non esistono istanze di questa classe non collegate a un ente.

Elenco proprietà:

1. **Nome** –stringa – Nome e cognome della persona correlata al ruolo.
2. **Ruolo** –stringa- indica la qualifica all'interno dell'ente cui è correlato. Il popolamento di questo campo avviene facendo riferimento in parte a una lista controllata (direttore, referente, sottosegretario, sindaco..) che consentirà di costruire elenchi
3. **Telefono** – stringa – E' una sequenza di numeri in formato descrittivo (non utile per la ricerca). Per i numeri di telefono internazionali, contiene anche il prefisso del paese.
4. **Fax** – stringa – E' una sequenza di numeri in formato descrittivo (non utile per la ricerca).
5. **E-mail** – stringa –
6. **UrlPaginaWeb** –stringa-
7. **Indirizzo** – stringa –
8. **Cap** – numero-
9. **Città** –stringa-
10. **Provincia** - codifica

Le proprietà telefono, fax, ecc... assumono valore solo se diversi dalle analoghe proprietà dell'Ente.

Classe EnteConSpecifica: E' una specializzazione, per sottoinsieme della classe Ente sui valori di *TipologiaIstituzionale*. Serve per specificare meglio la tipologia dell'Ente al fine di ottenere una ricerca più mirata. La specifica è correlata al valore assunto da *TipologiaIstituzionale*.

Elenco proprietà:

1. **Specifica** –codifica- Può assumere uno dei seguenti valori

TipologiaIstituzionale = ServizioCulturale

- **Biblioteca**
- **CentroDiDocumentazione**
- **CentroDiEducazioneAmbientale** – Centri di EA istituzionali
- **Museo**
- **OrtoBotanico/GiardinoZoologico.**

TipologiaIstituzionale = TerzoSettore

- **Associazione Ambientalista**
- **Enti Formativi**
- **AltreOrganizzazioni**

TipologiaIstituzionale = Agenzia/Authority

- **AgenziaLocale** (es. ARPA Toscana, Authority per il comune di Roma,...)
- **AgenziaNazionale** (es. ANPA, AIPA, Authority per la comunicazione,...)

Tipologia Istituzionale = Area Protetta

- Oasi
- Parco Nazionale
- Riserva Naturale Marina
- Altri Parchi

Tipologia Istituzionale = Area Servizi Di Pubblica Utilità

- Gestione Energia (es. Enel, Gas, Acqua,...)
- Gestione Rifiuti
- Telecomunicazioni
- Sicurezza & Prevenzione (es. Polizia, Corpo forestale dello stato,...)
- Trasporti
- Sanità (es. ASL RM2, Ospedale S. Matteo).

Tipologia Istituzionale = Enti Pubblici Centrali.

- Ministero Ambiente (Ministero, Servizio Sviluppo Sostenibile,...)
- Presidenza Del Consiglio (es. Dipartimento della protezione civile).
- Ministero Trasporti
-

Tipologia Istituzionale = Enti Pubblici Territoriali.

- Regionale (es. Assessorato regionale all'ambiente, Direzione Regionale per l'istruzione, ...)
- Provinciale (es. Assessorato provinciale alla cultura e istruzione).
- Comunale (es. Assessorato comunale cultura e istruzione).
- Interterritoriale (es. Comunità montana).
- Centro Di Orientamento
- Centro Di Formazione
- Camera Di Commercio.

Tipologia Istituzionale = Parti Sociali Ed Enti Di Loro Emanazione.

- Associazione Imprenditoriale (es. Federtrasporti....)
- Associazione Sindacale (es. CGIL,...)
- Associazione Professionale (es. CIDI).
- Associazione Di Categoria (es. Ordine degli architetti, ...).

Tipologia Istituzionale = Scuola.

- Scuola Dell'Infanzia
- Scuola Elementare
- Scuola Media
- Scuola Secondaria

Tipologia Istituzionale = Ente Universitario.

- Rettorato
- Facoltà
- Scuola Di Specializzazione
- Corso Di Perfezionamento
- Scuola Diretta Fini Speciali
- Dipartimento
- Istituto
- Centro interdipartimentale

Esempio: Università di Pisa. Un ente è l'Università stessa, che ha un suo indirizzo (il rettorato), dei riferimenti, una pagina Web, ecc. questo ente potrebbe comparire nella base di dati perché è un indirizzo utile. Un altro ente è la Facoltà di Scienze

MMFFNN che ha un suo indirizzo, i suoi riferimenti, ecc., questo ente è presente in quanto offre un corso di laurea in "Scienze Ambientali". Ecc. Dato un ente universitario, non necessariamente occorre inserire tutta la gerarchia di enti (es. se interessa l'ente ScuolaDiSpecializzazione, non necessariamente occorre creare anche gli enti a cui esso è subordinato), ma si può utilizzare la classe Riferimenti. Lo stesso discorso vale per la segreteria studenti, ove presente. La distinzione tra università pubblica e privata viene risolta lasciando a livello di interfaccia utente questa caratteristica e creando a livello fisico 2 tipologie: UniversitàPubblica e UniversitàPrivata.

Per chiarire ulteriormente:

Allo stato attuale delle cose, visto i dati che vengono al momento raccolti per i corsi universitari, è impensabile creare la gerarchia di enti universitari, così come descritta precedentemente, perché complicherebbe notevolmente l'immissione e la ricerca dei dati senza grossi benefici (infatti non stiamo facendo il sistema informativo per il MURST a cui potrebbe servire di avere un elenco di tutti gli enti ad esso correlato e magari con le loro offerte, che non sono solo nell'area ambientale). Per cui la soluzione sarebbe di non distinguere a livello di tipologia specifica dell'ente tra *Rettorato, Dipartimento,....*, ma di inserire solo l'ente *Università* tra i cui riferimenti posso avere gli indirizzi ed i ruoli che interessano INFEA, cioè solo gli indirizzi di quelle strutture che si occupano di ambiente (ancora una volta non immetteremo per esempio tutte le facoltà, i corsi di laurea, le scuole di specializzazione ecc.). Concludo con un esempio concreto. Se volessi inserire un *corso di laurea* svolto dal *dipartimento di biologia* della *facoltà* di scienze matematiche e naturali dell'*Università* di Pisa, inserirei un solo Ente (*l'Università* di Pisa), tra i cui riferimenti potrei avere l'indirizzo della *facoltà* di scienze matematiche (solo se interessa ed ho rilevato questi dati), l'indirizzo del *dipartimento di biologia* (solo se interessa ed ho rilevato questi dati), l'indirizzo della *segreteria studenti* (solo se interessa ed ho rilevato questi dati; in questo caso potrei averne una o più di segreteria studenti, es. una per il corso di laurea ed una per la scuola di specializzazione se il dipartimento di biologia fa un corso di laurea ed una scuola di specializzazione su tematiche ambientali), l'indirizzo della *direzione del dipartimento di biologia* (solo se interessa ed ho rilevato questi dati), etc.

Associazione Subordinato:

Questa associazione permette di stabilire delle gerarchie istituzionali tra enti (ad esempio tutte gli enti *Facoltà* sono subordinati all'ente *Università*). L'associazione è multipla, infatti un ente può essere subordinato a più di uno (es. CEA di Varese Ligure è subordinato a CREA di Genova e al WWF).

Associazione SedeLocale:

Questa associazione permette di risalire da un ente locale all'ente nazionale e viceversa conoscere tutte le sedi locali di un certo ente. (es. LIPU della regione Emilia è una sede locale della LIPU Nazionale).

Classe DocumentiIstitutivi: Per gli enti EA è utile avere la possibilità di consultare gli atti istitutivi. Per ogni ente EA possono esistere più documenti istitutivi.

Elenco proprietà:

1. **Documento** –*testo*- stralcio della legge o decreto istitutivo.

Classe FunzioneEA: Questa classe contiene le funzioni svolte da un certo ente..

Elenco proprietà:

1. **Funzione** –*codifica*- Può assumere uno dei seguenti valori:
 - **LaboratorioTerritoriale**
 - **CentroEsperienza**
 - **CoordinamentoRegionale**

Classe Servizio: indica i servizi offerti da un certo ente secondo una classificazione. Un servizio è offerto da uno o più enti, un ente può offrire uno o più servizi. Il servizio può essere fornito in una *Locazione* diversa da quella dell'ENTE, per cui esiste un'associazione opzionale con tale classe; se l'associazione non esiste per qualche istanza di servizio, vuol dire che la *Locazione* è quella dell'ENTE.

Elenco proprietà:

1. **TipologiaServizio** – *codifica* – assume uno dei seguenti valori
 - Consultazione,
 - Guida alla ricerca di materiali, informazioni e persone.
 - Supporto metodologico per la progettazione.
 - Facilitazioni logistiche (es.ecobus, alloggi ...)
 - Promozione informativa, formativa, educativa
 - Organizzazione di incontri (mostre, convegni, conferenze)
 - Assistenza per l'analisi e l'elaborazione. di dati e informazioni.
2. **Contatto:** - *testo* – Questa proprietà descrive le modalità per contattare gli organizzatori e può assumere il nome di una persona (ma non necessariamente) o di un Ente.
3. **Descrizione** – *testo* –Contiene informazioni generali tra cui anche gli orari e la modalità di fruizione del servizio, che non si ritiene di strutturare perciò non significativi per la ricerca e molto dipendente dal tipo di servizio e dall'ente che lo offre.
4. **Iscrizione** – *bool*- obbligatoria/opzionale.

Classe Locazione: Questa classe contiene i riferimenti specifici alla locazione di un oggetto del DB. Per semplicità di gestione si è definita un'associazione univoca con gli oggetti del DB, il che può comportare la duplicazione di questi dati per gli oggetti che insistono nella stessa specifica locazione.

Elenco proprietà:

1. **Stato** – *stringa* –.
2. **Area Geografica** – *stringa* - La necessità di esplicitare la sede nelle 4 proprietà (area geografica, regione, provincia e città) deriva dall'esigenza di voler ricercare gli enti in base ad un certo contesto territoriale. La proprietà può assumere i valori: *Italia centrale, Italia insulare, ecc.*
3. **Regione** – *stringa* -.
4. **Provincia** – *stringa* –
5. **Città** – *stringa* -
6. **Cap** – *numero* -.
7. **Indirizzo** – *stringa* -
8. **Collegamenti** – *testo* – descrive le modalità per raggiungere la località..
9. **Descrizione strutture** – *testo* - Descrizione delle strutture presenti. Non si è ritenuto di codificare le strutture (e quindi definire un'associazione) in quanto non significativa come condizione di ricerca.
10. **Barriere architettoniche** – *codifica* - (si, no) dice se la località è dotata di ausili architettonici per handicappati.

Classe OFFERTA: E' una classe che generalizza le proprietà dei vari *oggetti informativi* di INFEA, combinando gli oggetti informativi delle varie basi di dati (Esperienze in Andrea, Corsi di Formazione in Anfora, etc) e individuando le caratteristiche che le accomunano. Questa classe viene partizionata nelle due sottoclassi OffertaEA ed OffertaFormativa che descrivono essenzialmente i mondi derivati da Andrea e da Anfora con le loro evoluzioni. Si auspica la convergenza di questi mondi su altre associazioni, ad esempio quelle verso le classi Argomenti e Destinatari per le quali è necessaria una consistente fase di studio, che coinvolga specialisti della materia, non realizzabile nel presente progetto.

Elenco proprietà:

1. **Titolo** - *stringa* – Titolo dell'offerta. Tutte le entità informative prevedono una proprietà con queste caratteristiche. Le modalità di ricerca prevedono la selezione per sottostringa.
2. **Tipologia** – *codifica* – individua la tipologia di offerta (Offerta Formativa, Offerta Educativa). Questa proprietà è ridefinita dalle sue sottoclassi.
3. **Descrizione** – *testo* - descrizione libera dell'offerta. Su questa proprietà non è previsto di fare ricerca in base al contenuto.

Classe Offerta EA: Questa classe raggruppa le offerte EA, la cui fonte primaria è Andrea. Nella vecchia classificazione di Andrea c'era un po' di confusione tra Attività ed Esperienza. Alcune Attività di Andrea sono meglio definite nella classe Servizi (es. *Consulenza per la progettazione didattica*) altre nella classe Manifestazione (es. *Convegno, Concorso*). In Infea la classe *Proposta* con le sue specializzazioni cattura il concetto di una attività promossa (e/o coordinata, gestita) da un determinato Ente (nel senso che vengono messe a disposizione strutture o materiali, persone, ecc.) a cui possono partecipare altri gruppi o persone (es. Un percorso didattico promosso da Lega Ambiente). Nel momento in cui una *Proposta* dà vita ad una realizzazione (completata) questa diventa un'Esperienza documentata in Infea dalla sua Iperscheda (nota: lo stesso Ente che fa la Proposta può realizzare un'esperienza su di essa). Può esistere un'associazione tra Proposta ed Esperienza (es. tutte le esperienze riferite ad una determinata Proposta), ma questa associazione non è obbligatoria in quanto possono esistere Esperienze che non fanno riferimento ad alcuna Proposta o viceversa possono esistere delle proposte per cui non è presente nessuna Esperienza (cioè realizzazione).

Elenco proprietà:

1. **Tipologia - codifica** – Ridefinisce l'omonima proprietà di Offerta assumendo uno dei seguenti valori
 - *Proposta*
 - *Esperienza*
 - *Progetto*
 - *Manifestazione*

Nota: In caso di eliminazione di una Proposta (es. non è più attiva) le eventuali esperienze ad essa correlate possono rimanere nel database .

Classe Proposta/Esperienza: È una specializzazione della classe OffertaEA e quindi eredita da questa le associazioni con le classi, *Argomenti*, *DestinatariEA*, *Materiale* oltre alle associazioni con l'Ente che eredita dalla classe Offerta. Viene introdotta per comodità di lettura per via delle associazioni comuni (su *Finalità Educative* e *Contesto Ambientale*) che coinvolgono le sue sottoclassi.

Elenco proprietà sono quelle ereditate da Offerta:

1. **Tipologia – codifica-** ridefinisce i valori della proprietà ereditata in:
 - *Proposta*
 - *Esperienza*

Classe Proposta: È una specializzazione della classe Proposta/Esperienza e quindi eredita da questa le associazioni con le classi, *Contesto Ambientale*, *Argomenti*, *Finalità Educative*, *DestinatariEA*, *Materiale* oltre alle associazioni con l'*Ente* che eredita dalla classe *Offerta*.

Elenco proprietà sono quelle ereditate da *Offerta*:

1. **Tipologia** – *codifica*- ridefinisce i valori della proprietà ereditata in:
 - *Iniziativa*
 - *Visita/Soggiorno*
-

Classe Iniziativa: E una specializzazione della classe *Proposta*.

Elenco proprietà:

1. **Tipologia** – *codifica* – ridefinisce la proprietà di *Offerta* assumendo i seguenti valori:
 - *Percorso didattico*
 - *Progetto educativo*
 - *Città dei bambini*
 - *Ricerca educativa*
 - *Promozione*
 - *Aggiornamento* (es. corsi per insegnanti, etc)
 2. **Referente** – *stringa*- Persona a cui far riferimento.
 3. **PaginaWeb** –*url*- link all'eventuale pagina Web descrittiva dell'iniziativa.
-

Classe Visita/Soggiorno: E una specializzazione della classe *Proposta*. Oltre alle associazioni che eredita transitivamente dalla classe *Proposta*, contempla un'associazione sulla classe *Locazione* e *Contesto Geografico* che in generale possono essere diverse da quelle dell'*Ente*.

Elenco proprietà:

1. **Tipologia** – *codifica* – ridefinisce la proprietà di *Offerta* assumendo i seguenti valori:
 - *Visita guidata*
 - *Escursione*
 - *Campo scuola*
 - *Soggiorno*
 - *Soggiorno estivo*
2. **Durata** – *stringa*- E' espressa in forma testuale; non significativo per la ricerca.
3. **Contatto** – *testo* – Questa proprietà descrive le modalità per contattare gli organizzatori e può assumere il nome di una persona o di un Ente. La proprietà è necessaria in quanto l'Ente da contattare potrebbe non essere presente nel sistema, per cui occorre definire delle norme per l'assegnazione del valore onde evitare inutili e scomode ripetizioni.
4. **Iscrizione** – *codifica* – assume uno dei seguenti valori:
 - *Individuale*
 - *Gruppo*
5. **Accompagnatore** – *bool* – Specifica la necessità di accompagnatori (si/no).
6. **MaxPersone** – *intero* – Numero massimo di persone ammesse.

Classe Esperienza: È una specializzazione della classe Proposta/Esperienza e quindi eredita da questa le associazioni con le classi, *Contesto Ambientale*, *Argomenti*, *Finalità Educative*, *DestinatariEA*, *Materiale* oltre alle associazioni con l'*Ente* che eredita dalla classe *Offerta*. L'esperienza può essere la realizzazione di una Proposta, ma può esistere anche a prescindere da essa. L'esperienza compare nel sistema solo al suo completamento ed è obbligatoriamente descritta da un Iperscheda. Un'Esperienza, che nasce non correlata ad alcuna Proposta può dar luogo ad una Proposta (genera un'istanza della classe Proposta), qualora venga aperta alla partecipazione di altre persone o gruppi e vengono forniti ausili e /o supporto per la realizzazione di nuove Esperienze.

Elenco proprietà:

1. **Tipologia** – *codifica* – ridefinisce la proprietà di Offerta assumendo i seguenti valori (tutte le tipologie della classe Proposta):
 - *Campo scuola*
 - *Città dei bambini*
 - *Escursione*
 - *Soggiorno*
 - *Soggiorno estivo*
 - *Percorso didattico*
 - *Progetto educativo*
 - *Ricerca educativa*
 - *Promozione*
 - *Visita guidata*
2. **Referente** – *stringa* - Persona a cui far riferimento.
3. **PeriodoDiSvolgimento** – *anni* – E' una sequenza di anni non necessariamente contigua; occorre definire un formalismo per la ricerca con operatori temporali (< , >, dal al, negli anni).
4. **Iperscheda** – *url* - E' un link all'Iperscheda che descrive l'Esperienza.

Classe Progetto: Specializzazione della classe OffertaEA ottenuta per sottoinsieme sul valore *Progetto* della proprietà *Tipologia*. Descrive i Progetti in corso, di interesse del Ministero.

Elenco proprietà:

1. **Responsabile** –*stringa*- Nome e cognome del responsabile del progetto
 2. **Partecipanti** –*testo*- Descrive in forma testuale i partecipanti e i partner coinvolti nel progetto. Non si ritiene al momento di utilizzare questa proprietà per esprimere condizioni di ricerca.
 3. **DataAttivazione** –*data*- Data di inizio del progetto.
 4. **DurataInMesi** –*intero*- Durata prevista, espressa in mesi.
 5. **PaginaWeb** –*url*- link all'eventuale pagina Web descrittiva.
-

Classe Manifestazione: E' una specializzazione della classe *OffertaEA*. Le istanze di questa classe possono comparire nella classe *Agenda*. La classe ha un'associazione con *Argomenti*, *FontiFinanziare*, *Materiali*, *DestinatariEA* e *Locazione*, oltre alle associazioni con l'*Ente* che eredita dalla classe *Offerta*.

Elenco proprietà:

1. **Tipologia** – *stringa* – Ridefinisce l'omonima proprietà della classe *Offerta* assumendo uno dei seguenti valori:
 - Convegno/Conferenza
 - Mostra
 - Seminario
 - Campagna
 - Concorso
 - Settimana ambientale
2. **Contatto** – *testo* – Questa proprietà descrive le modalità per contattare gli organizzatori e può assumere il nome di una persona (ma non necessariamente) o di un Ente. La proprietà è necessaria in quanto l'Ente da contattare potrebbe non essere presente nel sistema, per cui occorre definire delle norme per l'assegnazione del valore onde evitare inutili e scomode ripetizioni.
3. **DataOraInizio** – *data* –
4. **DataOraFine** –*data*–
5. **PaginaWeb** –*url*– link all'eventuale pagina Web descrittiva.
6. **FileAllegato** –*url*– link all'eventuale file allegato (locandina, programma,...).

Classe SettimanaEA: E' una specializzazione della classe *Manifestazione* ottenuta per sottoinsieme sul valore *settimana ambientale* della proprietà *Tipologia*. La classe è introdotta per via dell'associazione con la classe *EventiSEA*.

Classe EventiSEA: Descrive gli eventi di una settimana ambientale. La descrizione di essi è semplificata rispetto a quella più generale dell'*Offerta* con tutte le sue associazioni. La classe ha un'associazione con *Locazione* ed una o più associazioni con *Ente* (gestisce, organizza, coordina, collabora), oltre che con *SettimanaEA*, occorre però che le istanze di queste classi siano ben individuabili per evitare l'accumulo nel tempo di oggetti scarsamente significativi. A livello di interfaccia per l'immissione e/o aggiornamento occorre semplificare la vista dell'utente (per es. eliminando alcuni dettagli di *Ente* e *Locazione*) per semplificarne l'utilizzo.

Elenco proprietà:

1. **Titolo** –*stringa*– Titolo dell'evento.
 2. **Contatto** –*stringa*– Persona o Ente da contattare.
 3. **Descrizione** –*testo*– Descrizione dettagliata dell'evento.
-

Classe Contesto Ambientale. Questa classe consente di analizzare l'ambiente di riferimento, come ecosistema, cui attiene l'Offerta e i Materiali EA e quindi di escludere offerte e materiali trattati senza correlazione all'ambiente di riferimento. La classe ha un'associazione multipla con *Offerta Ea e Materiale*.

Elenco proprietà:

1. **Tema - codifica** – assume uno dei seguenti valori:

- Acque dolci
- Ambiente antropizzato
- Ambiente litoraneo
- Ambiente marino
- Ambiente roccioso
- Ambiente urbano
- Bosco
- Prato
- Zone umide

Classe Finalità Educativa. Questa classe descrive le finalità educative degli interventi EA. La classe ha un'associazione multipla con *Offerta Ea e Materiale*.

Elenco proprietà:

1. **Descrittore - codifica** – assume uno dei seguenti valori:

- Conoscere i problemi dell'ambiente
- Conoscere l'ambiente
- Fare esperienze nell'ambiente
- Intervenire sull'ambiente
- Riflettere sull'educazione ambientale.

Classe Destinatari EA Questa classe indica i destinatari dell'*Offerta Educativa e/o dei Materiali*. È distinta dalla classe Destinatari Formazione in quanto meno articolata. Occorre prevedere la possibilità di ricerca incrociata (EA ed OFA) per le posizioni comuni, cioè *Giovani* (in cerca di prima occupazione in OFA), *adulti* (occupati, disoccupati e donne in OFA) e *tutti*.

Elenco proprietà:

1. **Posizione:** Può assumere uno dei seguenti valori:

- *Bambini*
 - *Giovani*
 - *Adulti*
 - *Anziani*
 - *Tutti*
-

Classe Argomenti: contiene una codifica degli argomenti oggetto dell'offerta EA. Viene lasciata distinta da *AreaTematica* di OFA poiché non esistono le possibilità di convergenza, all'interno del presente progetto.

Elenco proprietà:

- **Tema – codifica** – contempla la gestione del thesaurus di ANDREA.

Classe UrlArgomenti: i riferimenti alla descrizione degli argomenti, se presente..

Elenco proprietà:

- **Tema – codifica** – contempla la gestione del thesaurus di ANDREA.
- **UrlDescrizione –url-** Per alcuni argomenti ci può essere associato un file HTML che lo descrive.

Classe Richiesta Finanziamenti: Questa classe è una semplificazione della richiesta di finanziamenti, adottata per la registrazione e consultazione di richieste relative ad “Agende 21”. La gestione completa della procedura di finanziamenti del Ministero è di per sé molto più complessa, e, sebbene sia stata oggetto dello studio di fattibilità nel precedente contratto, è stato ritenuto di non includere la sua implementazione nel presente contratto di collaborazione. La soluzione adottata contempla comunque la possibilità di poter essere estesa per eventuali sviluppi futuri.

Elenco proprietà:

1. **Titolo progetto – stringa** –
2. **Acronimo – stringa** –
3. **Abstract – testo** –
4. **Protocollo arrivo – stringa** –
5. **Costo totale progetto - numero** –
6. **Quota cofinanziamento – numero** –
7. **Percentuale quota cofinanziamento su costo totale – numero** – è un campo calcolato dalle proprietà *Costo totale progetto* e *Quota cofinanziamento*
8. **Partner non finanziante – testo** -
9. **Tipologia di intervento – codifica** – questa codifica è specifica per Agende21, per generalizzarla occorre indagare. Assume uno o più tra i seguenti valori:
 - A
 - B
 - C
 - D
10. **Territorio coinvolto – codifica** – Codifica se possibile (esempio Regioni obiettivo 1) oppure stringa.
11. **Ammesso – Si/No**
12. **Note – testo-** Note o commenti sull'ammissibilità o sull'esclusione.

Classe Bando: Questa classe è una semplificazione della descrizione dei bandi di richieste finanziamento.

Elenco proprietà:

1. **Norma di riferimento** – *stringa* – Nome del bando con riferimento alla legge/decreto/circolare (esempio: Agende21, Poma, etc.).
2. **Valore del bando** – *numero* – ammontare del finanziamento disponibile per il bando.
3. **Quota massima finanziabile** – *numero* – Quota massima del finanziamento ammesso per ogni richiesta.
4. **Percentuale massima finanziamento** – *numero* – percentuale massima finanziabile rispetto al costo progetto.

Classe Criteri di ammissibilità: Questa classe rappresenta i criteri di ammissibilità specifici per “Agende 21”. La sua specializzazione, per il trattamento dei criteri relativi ad una richiesta correlata ad un qualsiasi bando, necessita di approfondimenti conoscitivi.

Elenco proprietà:

1. **Codice criterio** – *codifica* – Per Agende21 può assumere i seguenti valori:
 - Proponente >10000 abitanti
 - Somma dei proponenti >10000 abitanti
 - Proponente in regione ob. 1 e >8000 abitanti
 - Somma dei proponenti in regione ob. 1 e >8000 abitanti
 - Allegato documento di associazione/consorzio (solo per proponenti che siano associazioni o consorzi cioè con specifica tipologia “Interritoriali”)
 - Allegato copia del progetto
 - Allegato presenza bilanci preventivi.
2. **Rispondenza** – *si/no* – Per ogni criterio dice se è stato soddisfatto oppure no.

Classe Materiali: Descrive i materiali partendo dalla classificazione presente in ANDREA. La classe ha un'associazione multipla con *Autore* (obbligatoria) *Ambiente*, *Argomenti*, *Finalità*, *DestinatariEA*. L'associazione con l'*Ente* esprime il fatto che un certo materiale è stato *fornito* o *promosso* da un certo ente. L'associazione opzionale con *Offerta* esprime il fatto che un certo materiale è *offerto* dalla *Proposta* (es. i pannelli di mostra), oppure è stato *prodotto* da una determinata *Esperienza*, *Manifestazione*, ecc.. Nel caso di una richiesta sui materiali in cui compaiono anche gli Enti, occorre precisare che sono quelli che si ottengono tramite l'associazione diretta tra *Materiale* ed *Ente* e non tramite le associazione indirette ottenibili attraverso le classi *Proposta* e/o *Esperienza*.

Elenco proprietà:

1. **Titolo** –stringa-
2. **Tipologia** – *codifica* – Assume uno dei seguenti valori
 - *Audiovisivo* (Diapositiva, Musicassetta, VideoCassetta)
 - *Gioco*
 - *Ipermedia*
 - *Pannelli di mostra*
 - *Pubblicazione* (non è necessaria un'ulteriore specifica)
3. **Abstract** –*testo*- descrizione del materiale
4. **Lingua** – *codifica* – assume i valori: italiana, inglese, ecc.
5. **Editore** –stringa-
6. **AnnoDiEdizione** –anno-
7. **LuogoDiEdizione** –stringa-
8. **AltriRiferimentiBibliografici** – stringa –
9. **CollocatoInAndrea** –*bool*- si/no
10. **FileAllegato** –*url*- link all'eventuale file allegato per il downloading.

Nota: la specifica dettagliata delle proprietà comprese tra 5 e 8 è utile per poter ricostruire in fase di presentazione il Riferimento bibliografico del materiale secondo le norme di standardizzazione ISBD(m).

Classe Autore: Descrive gli autori di un materiale.

Elenco proprietà:

1. **Autore** – stringa-
-

Classe OffertaFormativa (Corsi): Comprende l'universo dei corsi di formazione così come inteso in ANFORA. Descrive le proprietà comuni a tutti i corsi. Questa classe viene partizionata, tramite il valore dell'attributo *Categoria* nelle varie classi relative ai differenti corsi (CF/I.S./Master, Università). La corrente versione non è quella contemplata in Anfora, ma la sua evoluzione. Eredita le proprietà dalla classe *Offerta*.

Elenco proprietà:

1. **Codice:** Identificativo univoco del progetto corsuale.
2. **Categoria:** Questa proprietà è stata introdotta per comodità di modellazione, discrimina tra i vari tipi di corso; può assumere uno dei seguenti valori:
 - *CF* Corso di formazione .
 - *I.S.* Intervento formativo della pubblica istruzione
 - *Università* Corso universitario.
 - **Master**
 - **IFTS**
3. **Durata:** Indica le annualità previste; si inserisce come corso annuale qualsiasi corso della durata inferiore all'anno solare, pertanto anche corsi che durano un solo giorno; può assumere i seguenti valori:
 - *annuale*
 - *biennale*
 - *triennale*
 - *quadriennale*
 - *quinquennale*
4. **Altre informazioni rilevanti** Informazioni rilevanti non contenute nei campi previsti (p.e. organizzazione del corso in collaborazione con altri enti o scuole di formazione; sponsorizzazioni e patrocini del corso, borse di studio per gli allievi...).

Classe Corso a distanza: Questa classe specifica le caratteristiche e le modalità di fruizione dei corsi a distanza.

Elenco proprietà:

1. **Organizzazione in moduli** -*bool* si/no
2. **Percentuale ore a distanza** - *numero* - indica la percentuale di ore di formazione a distanza, rispetto al totale.
3. **Percentuale ore in presenza** - *numero* - indica la percentuale di ore di formazione in presenza dell'insegnante, rispetto al totale.
4. **Possibilità di corsi individualizzati** -*bool*- si/no.
5. **Valutazione** -*testo*- può assumere uno o più dei seguenti valori (che vengono memorizzati in forma testuale in quanto non si ritengono significativi per la ricerca). I valori vengono comunque codificati in fase di inserimento sia per facilitare l'immissione che per uniformare la presentazione, secondo il seguente schema:
 - In itinere
 - In itinere con feed-back sui materiali
 - In itinere senza feed-back sui materiali
 - Finale
 - Nessuna valutazione

6. **Materiali didattici** *-testo-* può assumere uno o più dei seguenti valori (che vengono memorizzati in forma testuale in quanto non si ritengono significativi per la ricerca). I valori vengono comunque codificati in fase di inserimento sia per facilitare l'immissione che per uniformare la presentazione, secondo il seguente schema:
 - Materiale cartaceo
 - Materiale audiovisivo (videocassette, lezioni TV)
 - CD-Rom
 - Via Web
 7. **Servizi di supporto all'apprendimento** *-testo-* può assumere uno o più dei seguenti valori (che vengono memorizzati in forma testuale in quanto non si ritengono significativi per la ricerca). I valori vengono comunque codificati in fase di inserimento sia per facilitare l'immissione che per uniformare la presentazione, secondo il seguente schema:
 - Informativo/orientativo
 - Didattico
 - Tecnologico
 8. **Dotazione:** *-testo-* può assumere un insieme di valori che vengono memorizzati in forma testuale in quanto non significativi per la ricerca. I valori vengono comunque codificati in fase di inserimento sia per facilitare l'immissione che per uniformare la presentazione, secondo il seguente schema:
 - Videoregistratore
 - Fax
 - Computer
 - Stampante
 - Modem
 - Collegamento Internet
 - Altro (specificare).
-

Classe Interazione: Descrive le possibilità di interazione con docenti, tutor e partecipanti al corso..

Elenco proprietà:

1. **Modalità:** *-codifica-* può assumere uno dei seguenti valori
 - A distanza
 - In presenza
 2. **Strumento** *-testo-* può assumere un insieme di valori che vengono memorizzati in forma testuale in quanto non significativi per la ricerca. I valori vengono comunque codificati in fase di inserimento sia per facilitare l'immissione che per uniformare la presentazione, secondo il seguente schema:
 - Telefono
 - Fax
 - E-mail
 - Forum
 - Chat
 - Videoconferenza
 - Altro (specificare)
 3. **Soggetto** *-codifica-* può assumere uno dei seguenti valori
 - Docenti
 - Tutor
 - Altri partecipanti
-

Classe Corso CF/I.S./IFTS/Master: Comprende le proprietà comuni ai corsi IFTS, Master, corsi dei Centri di Formazione Professionali e della Pubblica Istruzione.

Elenco proprietà:

1. **Descrizione contenuto e finalità** descrizione sintetica dei contenuti e/o delle finalità.
2. **Attivato -bool-** Indica che il corso già programmato e finanziato (come lo sono tutti i corsi rilevati) è stato effettivamente attivato (non vuol dire iniziato), per cui sicuramente verrà svolto, anche se potrebbero cambiare i valori di qualche proprietà (ad esempio la data di svolgimento).
3. **Modalità temporali:** Indica la modalità temporale di svolgimento, ovvero il livello di impegno giornaliero. Può assumere uno dei seguenti valori:
 - *tempo pieno* (almeno 6 ore al giorno di formazione effettiva)
 - *tempo pieno residenziale*
 - *tempo parziale* (meno di 6 ore al giorno) (specificare..)
 - *fine settimana* (valore ammissibile solo per CF e Master non per I.S.)
 - *altra modalità* (specificare...).

Classe Corso CF/IFTS: Descrive le proprietà comuni ai corsi di Formazione ed ai corsi IFTS. Eredita tutte le proprietà dell'entità *Corsi CF/I.S/Master*.

Elenco proprietà:

1. **Ore complessive:** Indica le ore totali previste..
2. **Numero Massimo allievi ammessi:** Indica il numero massimo di partecipanti ammessi al corso.
3. **Quota di iscrizione:** -intero- Indica l'eventuale quota di iscrizione al corso, in lire IVA esclusa..
4. **Costo ora/allievo** -intero-
5. **Modalità di ammissione** – testo- Tipo di prova prevista per l'ammissione al corso (es. colloqui, prove scritte, test psico-attitudinali, ecc.). Non utile ai fini della ricerca, ma significativo a livello di restituzione dell'informazione, per cui è realizzato tramite una proprietà descrittiva.

Classe Corso CF: Descrive i corsi dei centri di formazione professionale. Eredita tutte le proprietà dell'entità *Corsi CF/IFTS*. (Corrisponde alla scheda di rilevazione dei corsi di formazione professionale)

Elenco proprietà:

1. **Natura del corso:** Indica una tra le seguenti tipologie:
 - *corso pubblico*, ovvero corso di formazione a finanziamento pubblico (incluso in un piano regionale)
 - *corso privato*
2. **Tipologia Formativa** Specifica la tipologia dell'attività formativa del corso e assume uno dei seguenti valori che determinano i destinatari del corso:
 - *corso di qualificazione*
 - *corso di specializzazione*
 - *corso di aggiornamento*
 - *corso di perfezionamento*

- *patente di mestiere*
3. **Tipo di gestione:** Classifica i progetti in una delle tipologie gestionali; il valore assunto da questo attributo è vincolato dal valore assunto dall'attributo *Natura del corso*. Può assumere i seguenti valori:
 - *gestione diretta/delegata* Interventi realizzati dalle regioni o dalle provincie in regime di delega.
 - *gestione convenzionata pubblica* corsi gestiti da enti pubblici (territoriali e no) convenzionati con l'Ente Regione
 - *gestione convenzionata privata* corsi gestiti da enti privati convenzionati con l'Ente Regione.
 - *gestione di mercato* corsi offerti su libero mercato a fine di lucro.. Questo valore può essere assunto solo dai corsi il cui valore dell'attributo *Natura del corso* è *corso privato*.
 - *altro* progetti non classificabili nelle tipologie relative ai corsi pubblici ed ai corsi privati (p.e. corsi gratuiti accessibili tramite borse di studio offerte da Ministeri, fondazioni ecc.).
 4. **Valutazione delle competenze:** Indica se è prevista una valutazione sugli esiti del corso. Può assumere i valori:
 - *si, in itinere*
 - *si, finale*
 - *no*.
 5. **Certificazione finale:** Indica la presenza o meno di una certificazione finale, classificandola tra le tipologie seguenti:
 - *frequenza* (riconosce la semplice partecipazione al corso)
 - *qualifica* (riconosce il possesso delle abilità tipiche di una determinata professione)
 - *specializzazione* (certificazione generalmente post qualifica, riconosce una specializzazione ulteriore nelle abilità tipiche della professione riconosciuta dalla qualifica).
 - *patente di mestiere* (riconosce la partecipazione ad un corso imposto per legge ai fini dell'esercizio di una professione (p.e. patentino per la conduzione di impianti termici).
 - *diploma* nel caso di Master.
 - *nessuno* (non è prevista certificazione finale)
 6. **Annualità in corso:** Prima, seconda, etc (assume valore solo per i corsi pluriennali).
 7. **Modalità didattica prevalente – codifica-** Indica la modalità didattica prevalente. Le modalità didattiche complete del corso sono espresse nella classe *SpecificDidattica*. Assume uno dei seguenti valori:
 - *in aula*
 - *in alternanza*
 - *a distanza*
 - *altro*.
-

Classe Corso IFTS: Descrive le proprietà dei corsi IFTS. Eredita tutte le proprietà e le associazioni dei *Corsi CF/IFTs*.

Elenco proprietà:

1. **Figura professionale** –*testo*- Descrive le possibili figure professionali prodotte del corso. Non significativo per la ricerca, in quanto non è possibile definire una codifica uniforme, ma può essere utile a livello di presentazione evidenziarne il contenuto.
 2. **Sbocchi occupazionali:** –*testo*- Descrive i possibili sbocchi professionali. Valgono le stesse considerazioni fatte per la proprietà precedente.
 3. **Agevolazioni finanziarie** –*testo*- Descrive le eventuali facilitazioni offerte (es. borse di studio, prestiti d'onore, ecc.). Valgono le considerazioni fatte per le due precedenti proprietà.
 4. **Valutazione dei crediti** – *codifica*- assume i seguenti valori:
 - valutazione iniziale, per l'accesso al corso
 - valutazione finale con riconoscimento certificato
 - valutazione finale senza riconoscimento certificato
 5. **Possibilità di piani individualizzati** –*bool*- si/no
 6. **Certificazione Intermedia** –*testo*- Contiene la descrizione della certificazione emessa.
 7. **Certificazione Finale** –*testo*- Contiene la descrizione della certificazione emessa.
 8. **Ore Complessive** – *numero* - Questa informazione non sempre è ricavabile dalla somma delle ore per semestre, in quanto non sempre viene fornito tale dettaglio.
 9. **Numero moduli** –*numero*- Numero totale dei moduli. Questa informazione non sempre è ricavabile dalla somma dei moduli per semestre, in quanto non sempre viene fornito tale dettaglio.
-

Classe ServiziSupporto IFTS: Contiene informazioni relative ai servizi di supporto per i corsi IFTS.

Elenco proprietà:

1. **Servizio**–*codifica*- assume i seguenti valori:
 - supporto informativo/orientativo,
 - psico-motivazionale,
 - professionale (tecnologico).
-

Classe Semestri IFTS: Contiene informazioni relative ai semestri dei corsi IFTS.

Elenco proprietà:

1. **Semestre** –*numero*- Un valore da 1 ad 8
 2. **Titolo** –*testo*- Titolo Relativo al semestre.
 3. **Ore** –*numero*- Ore previste per il semestre. Questa informazione non sempre è ricavabile dalla somma delle ore dei moduli per semestre, in quanto non sempre viene fornito tale dettaglio.
-

Classe Moduli IFTS: Descrive le proprietà dei corsi IFTS. Eredita tutte le proprietà e le associazioni dei *Corsi CF/IFTs*.

Elenco proprietà:

1. **Semestre** –*numero*- Un valore da 1 ad 8.
2. **Titolo del modulo** – testo-
3. **Dal** –*data*- Data Iniziale del modulo.
4. **Al** –*data*- Data Finale del modulo.
5. **Ore** –*numero*- Durata in ore del modulo.

Classe Master: Questa classe è un'estensione della classe Corsi di Formazione dal quale eredita tutte le proprietà (escluso tipo di gestione) e le associazioni. L'organizzazione in moduli viene espressa dall'associazione opzionale con la classe *Moduli*.

Elenco proprietà:

1. **Partecipazione a singoli moduli** –*bool* -: esprime la possibilità di partecipazione a singoli moduli del corso.
2. **Edizione** –*codifics*- prima, seconda,....
3. **Borse di studio** –*bool*- si/no
4. **Modalità fruizione borsa** – testo – descrive le modalità per usufruire di borsa di studio per i corsi che lo prevedono.

Classe Moduli: Questa classe descrive i moduli costituenti il corso Master. L'associazione col corso è opzionale. Possiede un'associazione con la classe *Periodo di svolgimento*.

Elenco proprietà:

1. **Titolo** –*stringa* -: titolo del modulo
2. **Ore** –*intero*- numero delle ore previste per il modulo

Classe Corso I.S.(Istruzione Secondaria): Questa entità descrive le attività formative svolte da Istituti Professionali di stato e Istituti Tecnici. Gli Istituti Professionali di stato rilasciano un attestato di qualifica al termine del primo biennio o triennio formativo, al termine del quinquennio rilasciano il diploma. L'entità eredita tutte le proprietà (attributi e associazioni) dall'entità *Corso* e dall'entità *Corsi CF/I.S.*. Nelle due schede cartacee proposte, l'unica differenza consiste nell'intestazione, in quanto una si riferisce a corsi di qualifica e l'altra a corsi di post qualifica, per questo entrambe le schede vengono modellate con la presente entità.

Elenco proprietà:

1. **Tipo:** Individua il tipo di corso; può assumere uno dei seguenti valori:
 - **Attestato di qualifica:** corsi di qualifica
 - **Diploma:** corsi di post qualifica

Classe Area Tematica: individuano gli argomenti trattati nei corsi OFA.

Elenco proprietà:

1. **Tema – codifica** – assume uno dei seguenti valori
 - acqua
 - agricoltura
 - aria
 - beni culturali e ambientali
 - conservazione della natura
 - ecologia
 - informazione ed educazione ambientale
 - energia
 - gestione e pianificazione
 - igiene e sanità
 - impatto ambientale
 - normativa ambientale
 - rifiuti
 - rumore
 - sicurezza del lavoro e dell’ambiente
 - qualità e certificazione
 - tecniche e tecnologie ambientali
 - territorio
 - turismo ambientale
 - urbanistica
 - verde urbano

Classe Fonti finanziarie: Questa entità contiene informazioni relative alle fonti di finanziamento dei corsi.

Elenco proprietà:

1. **Origine:** Può assumere uno dei seguenti valori:
 - **Fondi Comunitari**
 - **Fondi pubblici**
 - **Fondi privati**
 - **Altro**
2. **Specificità:** Il valore assunto da questo attributo dipende dal valore assunto da *Origine*.
 - **Origine = Fondi Comunitari**
 - *FSE, obiettivi, assi, etc* (specificare)
 - *Programmi e iniziative comunitarie* (specificare)
 - *Fondi comunitari di altra natura* (specificare)
 - **Origine = Fondi pubblici**
 - *Nazionali* (specificare)
 - *Regionali* (specificare)
 - *Provinciali* (specificare)
 - *Comunali* (specificare)
 - *Comunità Montana* (specificare)
 - **Origine = Fondi privati**
 - *Associazioni imprenditoriali*
 - *Associazioni di categoria*
 - *Autofinanziamento*
 - *Altro* (specificare)
 - **Origine = Altro**
 - *specificità*

Classe Destinatari Formazione Questa classe indica i destinatari dell'*Offerta Formativa*. Si tiene distinta dalla classe Destinatari dell'offerta EA in quanto diversamente articolata.

Elenco proprietà:

1. **Posizione:** Può assumere uno dei seguenti valori:
 - *Giovani in cerca di prima occupazione*
 - *Soggetti in situazione di svantaggio*
 - *Donne*
 - *Disoccupati*
 - *Occupati*
2. **Specificità:** La specificità della posizione è significativa essenzialmente per l'*Offerta Formativa*. Il valore dipende dal valore assunto dalla proprietà *Posizione*:
 - *Posizione = Giovani in cerca di I occupazione*
 - *drop-out*
 - *allievi*
 - *giovani con qualifica*
 - *giovani con diploma*
 - *universitari e neo laureati*
 - *Posizione = Soggetti in situazione di svantaggio*
 - *Portatori di handicap*
 - *Tossicodipendenti*
 - *Detenuti ed ex detenuti*
 - *Immigrati*
 - *Nomadi*
 - *Disoccupati adulti in età avanzata*
 - *Posizione = Disoccupati*
 - *Soggetti in riconversione*
 - *di lunga durata*
 - *mobilità*
 - *CIG*
 - *altro*
 - *Posizione = Occupati*
 - *apprendisti*
 - *contratto formazione lavoro*
 - *piccola e media impresa*
 - *grande impresa*
 - *istruzione e formazione professionale* (comprende i corsi di aggiornamento per insegnanti)
 - *pubblica amministrazione*
 - *altro*
 - *Posizione = Altro*
 - *specificità*

Classe *Periodo di svolgimento*: Indica il periodo e la sede di svolgimento del corso. Un Corso può avere più di una edizione in sedi e periodi differenti.

1. **IdLocazione** – codice.-
2. **IdCorso** – codice-
3. **Periodo.** –data- assume una delle seguenti forme.
 - *19nn/nm* (anno accademico)
 - *19/nn* (anno solare)
 - *dal/.. al ../..*

nella base di dati verrà memorizzato con opportuna convenzione, in modo da distinguere le precedenti forme, negli attributi:

- *dal* anno mese e giorno di inizio periodo
- *al* anno mese e giorno di fine periodo

Classe *Requisiti d'ingresso (Condizioni di Accesso)*: Descrive gli eventuali requisiti richiesti, per l'ammissione al Corso. Comprende sia il riquadro "Condizioni di accesso" che "Altre condizioni di accesso" della scheda cartacea..

Elenco proprietà:

1. **Titolo:** Può indicare il titolo di studio o altri requisiti culturali per l'accesso al corso, assume uno dei seguenti valori:
 - *nessuno* (l'accesso al corso non è vincolato ad alcun prerequisito)
 - *scuola dell'obbligo* (scolarità fino al compimento del 14° anno di età, qualunque sia il livello di studio raggiunto)
 - *licenza media* (diploma di scuola media inferiore)
 - *frequenza scuola secondaria superiore* (requisito culturale in itinere; l'allievo sta frequentando la scuola media superiore) (specificare)
 - *attestato di qualifica*
 - *diploma scuola secondaria* (diploma di scuola secondaria superiore) (specificare)
 - *diploma universitario*
 - *laurea* (diploma di laurea universitaria) (specificare)
 - conoscenza di una lingua straniera,
 - limite d'età
 - ..., ecc.

Classe *SpecificaDidattica*: Questa classe descrive le specifiche didattiche dei corsi di formazione. E' da notare che una data occorrenza dell'entità *SpecificaDidattica* può corrispondere a più progetti corsuali e viceversa., mentre la modalità didattica prevalente è una sola ed è espressa dall'omonima proprietà della classe *Corso di Formazione*.

Elenco proprietà:

1. **Modalità:** Indica la modalità didattica di realizzazione del corso, specificando una tipologia tra le seguenti:
 - *Lezioni in aula*
 - *Esercitazioni*
 - *Lezioni/esercitazioni a distanza*
 - *Lavoro in sottogruppo*
 - *Ricerca di gruppo*
 - *Discussione dei casi*
 - *Discussione guidata*
 - *Simulazione*
 - *Laboratori*
 - *Seminari*
 - *Visite guidate/viaggi di studio*
 - *Project work (progetti sul campo)*
 - *Stage*
 - *altro*
2. **Ore totali** –intero- numero di ore totali previste per ogni modalità didattica. Nota la scheda di rilevazione prevede questa possibilità solo per *stage*, ma potrebbe essere utile anche per tutte le altre modalità.

Classe *Docenza*: Indica la provenienza dei docenti oppure la tipologia di intervento che caratterizza la loro presenza all'interno del progetto formativo; è da notare l'associazione multipla bidirezionale con l'entità *Corsi CF/I.S/Master*.

Elenco proprietà:

1. **Tipo:** Indica la tipologia di intervento, può assumere uno dei seguenti valori.
 - *universitaria*
 - *aziendale*
 - *consulenza*
 - *testimonianza*
 - *tutoring*
 - *animatori*
 - *funzionari di P.A.*
 - *docenti interni*
 - *docenti iscritti all'albo regionale*
 - *altro* (specificare).

Classe Corso Università: comprende le proprietà comuni a tutti i corsi universitari. Esistono tre sottoinsiemi di questa entità determinati in base al valore assunto dall'attributo Tipologia e sono: Corso di Laurea, Diploma universitario e Specializzazione/Perfezionamento che hanno proprietà specifiche aggiuntive. Oltre alle associazioni che eredita dalla superclasse Offerta (es. l'ente universitario che lo gestisce) possiede due associazioni opzionali con la *Locazione* della segreteria studenti e con la locazione del coordinatore del corso.

Elenco proprietà:

1. **Tipologia:** classificazione del tipo di corso, può assumere uno tra i seguenti valori:
 - *Laurea*
 - *Diploma universitario*
 - *Scuola diretta a fini speciali*
 - *Scuola di specializzazione*
 - *Corso di perfezionamento*
2. **Anno di attivazione:** anno di attivazione del corso o della scuola
3. **Numero di esami/moduli:** numero di esami/moduli previsti durante il corso
4. **Test di ingresso:** specifica se è richiesto o meno un esame di ammissione
5. **Numero chiuso** –bool- si/no
6. **Numero posti** –intero- assume valore solo per i corsi a numero chiuso.
7. **Preside/coordinatore del corso.** –stringa- è il riferimento al responsabile del corso
8. **Referente** – stringa – è il referente del corso.

Classe Corso di Laurea: è una specializzazione dell'entità Corso Università ottenuta come sottoinsieme sul valore dell'attributo Tipologia per il valore Laurea.

Elenco proprietà:

1. **Indirizzo:** indica l'indirizzo del corso di laurea.

Classe Diploma Universitario: è una specializzazione della classe Corso Università ottenuta come sottoinsieme su Tipologia per il valore *Diploma universitario*.

Elenco proprietà:

1. **Orientamento:** indica l'orientamento del corso di laurea.

Classe Specializzazione/Perfezionamento: è una specializzazione dell'entità Corso Università ottenuta come sottoinsieme su Tipologia per i valori Scuola di Specializzazione o Corso di Perfezionamento.

Elenco proprietà:

1. **RequisitiDiIngresso:** requisiti per l'ammissione al corso. I valori saranno immessi in forma testuale in quanto non è richiesto la ricerca di un corso universitario in base ai requisiti di ingresso.
2. **Materie** – testo- Elenco delle materie svolte, in forma descrittiva.
3. **QuotaDiIscrizione** –numero-

Classe Perfezionamento: è una specializzazione dell'entità Specializzazione/Perfezionamento ottenuta come sottoinsieme sul valore dell'attributo Tipologia per il valore Corso di Perfezionamento.

Elenco proprietà:

1. **Periodo di svolgimento:** Data nel formato dal... al
-

Classe Agenda: Questa classe è utile per poter segnalare eventi che hanno un particolare rilievo. Essenzialmente serve per gestire lo strumento Agenda nelle modalità previste dal disegno del sistema informativo. La popolazione di questa classe avviene sotto il controllo di un ActivityManager che seleziona le offerte per esse significative. Gli esemplari presenti in questa classe sono elementi delle classi Manifestazione (Conferenza/Convegno, Mostra, Campagna, Concorso, Seminario), e Segnalazioni.

Elenco proprietà:

1. **NomeTabella** *-stringa-* Individua la classe che descrive l'Offerta da segnalare in Agenda
2. **DataScadenza** *-data-* Data ultima di permanenza dell'evento in Agenda.

Classe Segnalazione: Questa classe descrive eventi da segnalare in Agenda, che non sono classificabili come Offerta con tutte le sue specializzazioni ed associazioni (es. Il ministro inaugura un LT in una certa data). Lo scopo è solo legato alla permanenza delle sue istanze in Agenda.

Elenco proprietà:

1. **Titolo:** *-stringa-*
2. **Descrizione** *- testo-*
3. **DataInizio** *-data-*
4. **DataFine** *-data-*
5. **PaginaWeb** *-url-* link all'eventuale pagina Web descrittiva.
6. **FileAllegato** *-url-* link all'eventuale file allegato (locandina, programma,...).

Classe Riunione: Questa classe contiene informazioni relative alle riunioni di interesse per gli utenti Infea. Sostanzialmente gli oggetti di questa classe sono legati alla loro permanenza in Agenda. L'utilizzo previsto è essenzialmente quello di segnalare riunioni, relative a progetti promossi dal Ministero, agli utenti indicati nella proposta di sistema informativo come utenti Labnet (cioè Laboratori Territoriali, Centri EA, ecc.).

Elenco proprietà:

1. **Tema:** *-stringa-* Breve titolo della riunione.
 2. **Descrizione** *- testo-* Descrizione dei contenuti e degli obiettivi della riunione. Non si prevede di esprimere condizioni di ricerca su questa proprietà.
 3. **Partecipanti** *-testo-* Elenco dei partecipanti in forma testuale. Non si prevede di esprimere condizioni di ricerca su questa proprietà.
 4. **Luogo** *-stringa-* Luogo della riunione in forma testuale. Non si prevede di esprimere condizioni di ricerca su questa proprietà.
 5. **DataOraInizio** *-data-* Data e ora di inizio.
 6. **DataOraFine** *-data-* Data e ora fine
-

Classe UtenteDelSistema: Questa classe contiene informazioni sugli utenti del sistema informativo Infea. L'utilizzo di questa classe è legata alle funzioni di amministratore del sistema. Per maggiori informazioni su Utenti e Ruoli del sistema informativo si veda il documento relativo a tale proposta. Un utente può appartenere a più gruppi e quindi possedere tutte le autorizzazioni dei vari gruppi a cui appartiene. Per gli utenti generici (utenti saltuari via Web) non occorre definire un identificativo né una password, in quanto accedono al sistema in modalità Ospite e sono autorizzati solo alle funzionalità previste per questo ruolo.

Elenco proprietà:

1. **Utente:** *-stringa-* Identificativo dell'Utente
2. **NomeEsteso** *-stringa-* Nome Esteso dell'utente ad utilizzo dell'amministratore.
3. **Password** *-stringa-* Password di accesso al sistema Infea.
4. **Descrizione** *-stringa-* Breve descrizione ad uso dell'amministratore del sistema.

Classe Gruppo: Questa classe contiene informazioni sui gruppi di utenti presenti nel sistema. Il gruppo è l'oggetto su cui si basa il meccanismo delle autorizzazioni di accesso al sistema Infea. Esistono dei gruppi predefiniti che rientrano nella classificazione presente nella proposta di sistema informativo (es. Guest, SystemAdministrator, ecc.). L'amministratore del sistema ha la possibilità di definire nuovi gruppi o di rimuoverne.

Elenco proprietà:

1. **NomeGruppo:** *-stringa-* Identificativo del gruppo
2. **Descrizione** *-stringa-* Breve descrizione ad uso dell'amministratore del sistema.

Classe Autorizzazione: Questa classe contiene informazioni sulle autorizzazioni assegnate ad ogni gruppo di utenti presenti nel sistema. L'amministratore del sistema ha la possibilità di concedere e/o rimuovere autorizzazioni.

Elenco proprietà:

1. **NomeGruppo:** *-stringa-* Identificativo del gruppo a cui si concede l'autorizzazione
2. **Oggetto** *-codifica-* Componente del sistema Infea su cui si concede l'autorizzazione espressa dalla successiva proprietà.
3. **Autorizzazione** *-codifica-* Tipo di autorizzazione concessa.

Esempio: L'istanza (Guest,Agenda,SolaLettura) indica che gli utenti generici del sistema (appartenenti al gruppo Guest) hanno accesso in lettura agli eventi presenti in Agenda, ma non possono inserirne di nuovi né modificare gli eventi esistenti.

Le seguenti classi sono evidenziate per segnalare i servizi ad essi associati. Questi servizi consentono di generare dinamicamente elenchi di URL e liste di Distribuzione, selezionando, mediante filtri, oggetti dalla classe *Riferimento*. Qualora si voglia rendere permanente l'insieme di oggetti ottenuto, ad esso viene assegnato un nome e memorizzato nella opportuna classe (ElencoDiURL o ListeDiDistribuzione):

Classe ElencoDiURL: Questa classe descrive gli elenchi di URL permanenti

Elenco proprietà:

1. **Nome:** *-stringa-* Nome dell'elenco permanente
-

Classe ListeDiDistribuzione:

Elenco proprietà:

1. **Nome:** *-stringa-* Nome della lista di distribuzione permanente
-

2. Progettazione logica e fisica della base di dati

In questo capitolo è descritta la fase di progettazione logica e fisica della base di dati. Nel seguito viene descritta la trasformazione dello schema concettuale, di Fig. 1 nel modello dei dati relazionale. Per l'implementazione dello schema della base di dati è stato adottato il linguaggio SQL, perché esiste una definizione di standard da parte di organizzazioni internazionali (ANSI e ISO) e perché largamente adottato da molti sistemi di gestione di base di dati disponibili commercialmente. Questa scelta garantisce una buona portabilità in differenti ambienti senza generare dipendenze da specifici prodotti hardware e software.

Norme adottate nella realizzazione della base di dati.

1. Tutte le proprietà dello schema concettuale definite col tipo *codifica* sono state rappresentate nelle relazioni risultanti tramite valori numerici univoci. Nelle relazioni i nomi di tali proprietà hanno come prefisso la stringa *Cod*. La corrispondenza tra valore numerico e forma presentabile della proprietà (es. *10 = Dipartimento Ministeriale*) è mantenuta tramite un'unica relazione (la relazione *Codifica*). Questa scelta è stata adottata principalmente in funzione di una possibile variazione delle forme presentabili, in quanto vengono localizzati i possibili interventi di modifica (es. basta sostituire la relazione *Codifica* per avere la versione della base di dati in un'altra lingua). Inoltre tale soluzione consente di ridurre i tempi di trasferimento dei dati tra il Server della base di dati e le applicazioni in quanto le singole relazioni contenenti i valori codificati sono molto più compatte e infine consente di ridurre al minimo la ridondanza e quindi l'occupazione di spazio; tutte e tre le motivazioni determinano una maggiore efficienza del sistema. La scelta non penalizza l'usabilità, in quanto l'accesso alla base di dati non avverrà mai direttamente tramite il linguaggio nativo dello specifico DBMS, ma sempre tramite interfacce utenti che assumono il carico di trasformazione dei valori codificati internamente in valori presentati all'utente finale.
2. Tutti gli identificatori delle relazioni hanno un nome prefissato da *Id* e sono stati definiti di tipo stringa di 8 caratteri. I valori per tali proprietà consistono in un numero progressivo di 6 o 7 cifre prefissati da uno o due lettere che richiamano il nome dell'oggetto che identificano (es. EU000025 è l'identificatore di un Ente Universitario).
3. Tutte le proprietà di codifica (attributi il cui nome inizia con *Cod*) sono definiti sul tipo SQL *smallint*.
4. In alcune relazioni particolarmente importanti dal punto di vista implementativo sono state aggiunte, essenzialmente per problemi gestionali, le seguenti proprietà:
 - *DataPrimoInserimento* data e ora del primo inserimento della specifica istanza nella base di dati.

- *DataUltimaModifica* data e ora dell'ultima modifica apportata ai valori della specifica istanza.
- *UtentePrimoInserimento* Identificativo dell'utente che ha sottoposto l'inserimento della specifica istanza.
- *UtenteProponente* Identificativo dell'utente che ha sottoposto l'ultima modifica della specifica istanza.
- *UtenteApprovante* Identificativo dell'amministratore che ha approvato l'inserimento, o l'ultima modifica della specifica istanza.
- *NoteAmministratore* attributo testuale per eventuali note ad uso dell'amministratore.

Queste proprietà sono accessibili solamente da utenti con ruolo di amministratore.

2.1 Trasformazione dello schema concettuale

Il primo passo della traduzione dal modello concettuale al modello relazionale è consistito nella risoluzione delle gerarchie IS-a.

2.1.1 Trasformazione della classe Ente

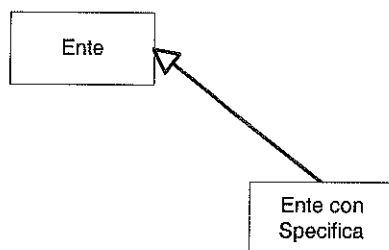


Figura 1. Gerarchia Enti dello schema concettuale

La gerarchia di Figura 1 è stata risolta, per ottimizzazione implementativa, definendo un'unica relazione *Enti*.

Di seguito compare la definizione della relazione corrispondente espressa in linguaggio SQL.

Relazione Enti

```
CREATE TABLE Enti (  
    IdEnte char (8) NOT NULL ,  
    Denominazione varchar (255) NOT NULL ,  
    CodTipologiaIstituzionale smallint NOT NULL ,  
    CodSpecificaTipologia smallint NULL ,  
    Privato char (1) NULL ,  
    SitoWeb varchar (255) NULL ,  
    Descrizione text NULL ,  
    Logo image NULL ,  
    IdEnteSedeNazionale char (8) NULL ,  
    IdEnteSovraordinante char (8) NULL ,  
    DataPrimoInserimento datetime NOT NULL ,  
    DataUltimaModifica datetime NOT NULL ,  
    UtentePrimoInserimento varchar (20) NOT NULL ,  
    UtenteProponente varchar (20) NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Ente PRIMARY KEY CLUSTERED (  
        IdEnte  
    )  
)
```

La tipologia di ente viene discriminata dall'attributo *CodTipologiaIstituzionale* che può assumere uno dei seguenti valori:

CodTipologiaIstituzionale	
01	Ente Pubblico Centrale
02	Ente Pubblico Territoriale
03	Ente Universitario
04	Servizio Culturale
05	Scuola
06	Ente o Istituto di Ricerca pubblico o privato
07	Area Servizio di pubblica utilità
08	Parte Sociale o enti di loro emanazione
09	Area Protetta
10	Terzo Settore
11	Organismo internazionale
12	Impresa
13	Agenzia/Authority

Privato segnala la natura dell'Ente, assume valore binario (vero, falso).

CodSpecificaTipologia assume valori che dipendono dal valore assunto da

CodTipologiaIstituzionale della relazione *Enti*:

CodTipologiaIstituzionale = EntePubblicoCentrale →

CodSpecificaTipologia	
01	Ministero economia e finanze
02	Ministero ambiente e tutela del territorio.
03	Presidenza Del Consiglio
04	Ministero affari esteri
05	Ministero interno
06	Ministero giustizia
07	Ministero attività produttive
08	Ministero istruzione, università e ricerca scientifica
09	Ministero lavoro e politiche sociali
10	Ministero difesa
11	Ministero politiche agricole e forestali
12	Ministero infrastrutture e trasporti
13	Ministero salute
14	Ministero beni culturali
15	Ministero comunicazioni

CodTipologiaIstituzionale = EntePubblicoTerritoriale →

CodSpecificaTipologia	
01	Regionale
02	Provinciale
03	Comunale
04	Interterritoriale
05	Centro di orientamento
06	Centro di formazione
07	Camera di commercio

CodTipologiaIstituzionale = EnteUniversitario →

CodSpecificaTipologia	
01	Rettorato
02	Facoltà
03	Scuola di specializzazione
04	Scuola di perfezionamento
05	Dipartimento
06	Istituto
07	ScuolaDirettaFiniSpeciali
08	Centro interdipartimentale

Dato un ente universitario, non necessariamente occorre inserire tutta la gerarchia di enti (es. se interessa l'ente ScuolaDiSpecializzazione, non necessariamente occorre creare anche gli enti a cui esso è subordinato), ma si può utilizzare la classe Riferimenti. Lo stesso discorso vale per la segreteria studenti, ove presente. La distinzione tra università pubblica e privata viene risolta lasciando a livello di interfaccia utente questa caratteristica e creando a livello fisico 2 tipologie: UniversitàPubblica e UniversitàPrivata che verranno gestite opportunamente, in maniera trasparente per l'utente.

CodTipologiaIstituzionale = AreaProtetta →

CodSpecificaTipologia	
01	Oasi
02	Parco nazionale
03	Riserva naturale marina
04	Altri parchi

CodTipologiaIstituzionale = Scuola →

CodSpecificaTipologia	
01	Scuola dell'infanzia
02	Scuola elementare
03	Scuola media
04	Scuola secondaria

CodTipologiaIstituzionale = ServizioCulturale →

CodSpecificaTipologia	
01	Biblioteca
02	Centro di Documentazione
03	Centro di Educazione Ambientale
04	Museo
05	Orto botanico/giardino zoologico

CodTipologiaIstituzionale = AreaServiziDiPubblicaUtilità →

CodSpecificaTipologia	
01	Gestione energia (Enel,gas,...)
02	Gestione rifiuti
03	Telecomunicazioni
04	Sicurezza e Prevenzione
05	Trasporti
06	Sanità

CodTipologiaIstituzionale = TerzoSettore →

CodSpecificaTipologia	
01	Associazione Ambientalista
02	Enti Formativi
100	Altre Organizzazioni

CodTipologiaIstituzionale = Agenzia/Authority →

CodSpecificaTipologia	
01	Locale
02	Nazionale

CodTipologiaIstituzionale = PartiSocialiEdEntiDiLoroEmanazione →

CodSpecifica Tipologia	
01	Associazione imprenditoriale
02	Associazione sindacale
03	Associazione professionale
04	Associazione di categoria

2.1.2 Trasformazione delle associazioni che coinvolgono la classe Ente

Le seguenti associazioni dell'entità Ente con se stesso (Figura 3) sono state modellate aggiungendo due attributi alla relazione Enti (*IdEnteSedeNazionale* e *IdEnteSovraordinante*)

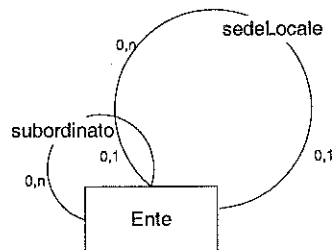


Figura 3. Associazioni di Ente con se stesso

Tutte le associazioni fra *Ente* e *Offerta* (Figura 4) vengono modellate tramite la relazione *OffertaEnte*, in cui l'*IdOfferta* è relativa all'*Id* di una sottoclasse di *OffertaEa* o di *OffertaFormativa*.

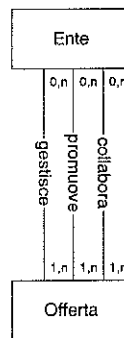


Figura 4.

Relazione OffertaEnte

```
CREATE TABLE OffertaEnte (
    IdOfferta char (8) NOT NULL ,
    IdEnte char (8) NOT NULL ,
    CodTipoAssociazione smallint NOT NULL ,
    CONSTRAINT PK_OffertaEnte PRIMARY KEY CLUSTERED (
        IdEnte,
        IdOfferta,
        CodTipoAssociazione
    ),
    CONSTRAINT FK_OffertaEnte FOREIGN KEY (
        IdEnte
    ) REFERENCES dbo.Enti (
        IdEnte
    ))
```

La relazione modella l'associazione multipla bidirezionale tra *Offerta* ed *Ente*.

La tipologia della associazione viene discriminata dall'attributo *CodTipoAssociazione* che può assumere uno dei seguenti valori:

CodTipoAssociazione	
01	Collabora
02	Promuove
03	Gestisce



Figura 4a. Associazione Ente/Riferimento

L'associazione (1:n) tra la classe *Ente* e la classe *Riferimento* dello schema concettuale viene realizzata tramite la seguente relazione.

```

CREATE TABLE Riferimenti (
  IdEnte char (8) NOT NULL,
  NumProgressivo smallint NOT NULL,
  Ruolo varchar (255) NOT NULL ,
  Nome varchar (255) NULL,
  Telefono varchar (255) NULL,
  Fax varchar (255) NULL,
  Email varchar (255) NULL,
  UrlPaginaWeb varchar (255) NULL,
  Indirizzo varchar (255) NULL,
  Cap integer NULL,
  Citta varchar (255) NULL,
  SiglaProvincia char (2) NULL,
  CONSTRAINT PK_Riferimenti PRIMARY KEY CLUSTERED (
    IdEnte,
    NumProgressivo
  ), CONSTRAINT FK_Riferimenti FOREIGN KEY
  (
    IdEnte
  )REFERENCES dbo.Enti (
    IdEnte
  )
)
  
```

I valori di *Indirizzo*, *Cap*, *Citta*, *SiglaProvincia* sono immessi solo se diversi da quelli della locazione dell'ente.

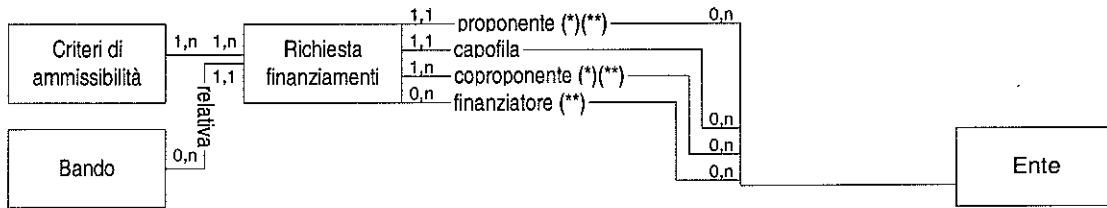


Figura 4b. Associazione Ente/Richiesta finanziamenti.

```

CREATE TABLE Bando(
  IdBando char (8) NOT NULL,
  CodNormaDiRiferimento smallint NOT NULL,
  ValoreDelBando int,
  CodValuta smallint,
  QuotaMaxFinanziabile int,
  PercentualeMaxFinanziamento int,
  CONSTRAINT PK_Bando PRIMARY KEY CLUSTERED (
    IdBando
  )
)
  
```

CodNormaDiRiferimento	
01	Agende 21
02	Poma
03	..

CodValuta	
01	Lire
02	Euro
03	..

```

CREATE TABLE RichiesteFinanziamento (
  IdRichiesta char (8) NOT NULL,
  IdBando char (8) NOT NULL,
  TitoloProgetto varchar (255) NOT NULL ,
  Acronimo char (30) NOT NULL ,
  Abstract text NULL,
  ProtocolloArrivo varchar (255) NOT NULL ,
  CostoTotaleProgetto int NULL,
  CodValuta smallint,
  QuotaCofinanziamento int NULL,
  PartnerNonFinanziante text NULL,
  TipologiaIntervento char (12),
  CodTerritorioCoinvolto smallint,
  IdCapofila char (8) NOT NULL,
  IdProponente char (8) NOT NULL,
  DataAalboorgProponente datetime NULL ,
  QuotaFinanziataProponente int NULL,
  Ammesso char(1),
  Note text NULL,
  CONSTRAINT PK_RichiesteFinanziamento PRIMARY KEY CLUSTERED (
    IdRichiesta
  ), CONSTRAINT FK_RichiesteFinanziamento FOREIGN KEY
  (
    IdBando
  )REFERENCES dbo.Bando (
    IdBando
  )
)
)

```

Il campo **TipologiaIntervento** contiene una stringa di valori, tra i seguenti, separati da virgola. Tale soluzione è stata adottata per facilità di implementazione, essendo questo requisito mutato dopo la fase di progettazione ed implementazione (all'inizio poteva assumere un singolo valore).

tipologiaIntervento	
A	
B	
C	
D	

CodTerritorioCoinvolto	
01	Regioni Obiettivo 1

```

CREATE TABLE CriteriAmmissibilita (
  IdRichiesta char (8) NOT NULL,
  CodCriterio smallint NOT NULL ,
  Rispondenza char (1) NOT NULL ,
  CONSTRAINT PK_CriteriFinanziamento PRIMARY KEY CLUSTERED (
    IdRichiesta, CodCriterio
  ), CONSTRAINT FK_CriteriFinanziamento FOREIGN KEY (
    IdRichiesta
  ) REFERENCES dbo.RichiesteFinanziamento (
    IdRichiesta
  )
)

```

CodCriterio (x Agende 21)	
01	Proponente >10000 abitanti
02	Somma dei proponenti >10000 abitanti
03	Proponente in regione ob. 1 e >8000 abitanti
04	Somma dei proponenti in regione ob. 1 e >8000 abitanti
05	Documento di associazione (solo per consorzi e associazioni)
06	Copia del progetto
07	Bilanci preventivi

```

CREATE TABLE CoProponentiRichiesta (
  IdRichiesta char (8) NOT NULL,
  IdEnte char (8) NOT NULL,
  DataAalboorg datetime NULL ,
  QuotaFinanziata int NULL,
  CONSTRAINT PK_CoProponentiRichiesta PRIMARY KEY CLUSTERED (
    IdRichiesta, IdEnte
  ), CONSTRAINT FK_CoProponentiRichiesta FOREIGN KEY (
    IdRichiesta
  ) REFERENCES dbo.RichiesteFinanziamento (
    IdRichiesta
  )
)

```

```

CREATE TABLE CoFinanziatoriRichiesta (
  IdRichiesta char (8) NOT NULL,
  IdEnte char (8) NOT NULL,
  QuotaFinanziata int NULL,
  CONSTRAINT PK_CoFinanziatoriRichiesta PRIMARY KEY CLUSTERED (
    IdRichiesta, IdEnte
  ), CONSTRAINT FK_CoFinanziatoriRichiesta FOREIGN KEY (
    IdRichiesta
  ) REFERENCES dbo.RichiesteFinanziamento (
    IdRichiesta
  )
)

```

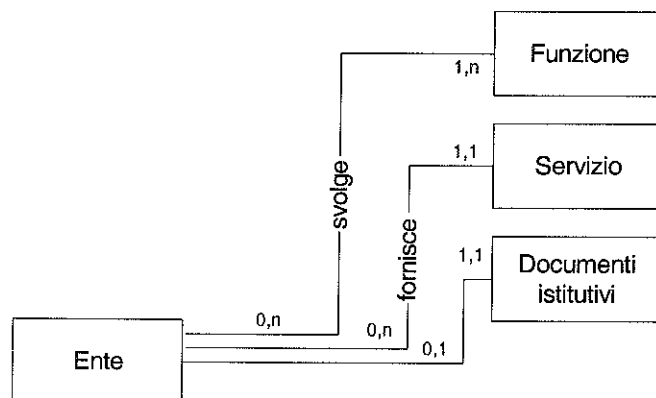


Figura 5.

Le classi *Funzione*, *Servizio*, *Documenti istitutivi* e le loro associazioni con la classe *Ente* (Figura 5) sono state realizzate tramite le relazioni *FunzioniEA*, *Servizi*, *DocumentiIstitutivi* (quest'ultima per eventuali scopi futuri, infatti l'associazione potrebbe diventare del tipo 1:n qualora si ritenesse di associare i documenti istitutivi reali per esempio in forma di file distinti).

Relazione FunzioniEA

```

CREATE TABLE FunzioniEA (
  IdEnte char (8) NOT NULL ,
  CodFunzione smallint NOT NULL ,
  CONSTRAINT PK_FunzioneEa PRIMARY KEY CLUSTERED
  (
    IdEnte,
    CodFunzione
  ),
  CONSTRAINT FK_FunzioneEa FOREIGN KEY
  (
    IdEnte
  ) REFERENCES dbo.Enti (
    IdEnte
  )
)

```

CodFunzione	
01	Laboratorio territoriale
02	Centro esperienza
03	Coordinamento Regionale

Relazione Servizi

```
CREATE TABLE Servizi (  
    IdServizio char (8) NOT NULL ,  
    IdEnte char (8) NOT NULL ,  
    CodTipologiaServizio smallint NOT NULL ,  
    Contatto varchar (255) NULL ,  
    Descrizione text NULL ,  
    Iscrizione char (1) NOT NULL ,  
    IdLocazione char (8) NOT NULL ,  
    DataPrimoInserimento datetime NOT NULL ,  
    DataUltimaModifica datetime NOT NULL ,  
    UtentePrimoInserimento varchar (20) NOT NULL ,  
    UtenteProponente varchar (20) NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Servizi PRIMARY KEY CLUSTERED (  
        IdServizio  
    ),  
    CONSTRAINT FK_Servizi FOREIGN KEY (  
        IdEnte  
    ) REFERENCES dbo.Enti (  
        IdEnte  
    )  
)
```

Questa relazione rappresenta i Servizi e l'associazione (1:n). con gli *Enti* tramite l'attributo *IdEnte*.

CodTipologiaServizio	
01	Consultazione,
02	Guida alla ricerca di materiali, informazioni.
03	Supporto metodologico per la progettazione
04	Facilitazioni logistiche
05	Promozione informativa
06	Organizzazione di incontri (mostre, convegni, conferenze)
07	Assistenza per l'analisi e l'elaborazione. di dati e informazioni.

La proprietà *IdLocazione* rappresenta l'associazione tra la classe *Servizio* e la classe *Locazione*. dello schema concettuale, il valore assunto è uguale al valore di *IdServizio* o di *IdEnte* a seconda che la locazione del servizio coincide con quella dell'ente oppure no.

La proprietà *IdEnte* rappresenta l'associazione "fornisce" tra la classe *Ente* e la classe *Servizio*. dello schema concettuale.

Relazione DocumentiIstitutivi

```
CREATE TABLE DocumentiIstitutivi (  
    IdEnte char (8) NOT NULL ,  
    IdDocumento char (8) NULL ,  
    Documento text NOT NULL ,  
    CONSTRAINT PK_DocumentiIstitutivi PRIMARY KEY CLUSTERED  
    (  
        IdEnte  
    ),  
    CONSTRAINT FK_DocumentiIstitutivi FOREIGN KEY  
    (  
        IdEnte  
    ) REFERENCES dbo.Enti (  
        IdEnte  
    )  
)
```

Questa relazione stralci di documenti relativi agli aspetti giuridici dell'Ente. Come già detto, questa relazione viene tenuta per eventuali usi futuri (cioè migrazione verso l'associazione 1:n). Per il momento, data l'associazione 1:1 non sarebbe necessaria la proprietà *IdDocumento*, che viene tuttavia mantenuta per usi futuri (N.B. in caso di associazione 1:n *IdDocumento* dovrà far parte della chiave primaria).

2.1.3 Trasformazione della classe Offerta

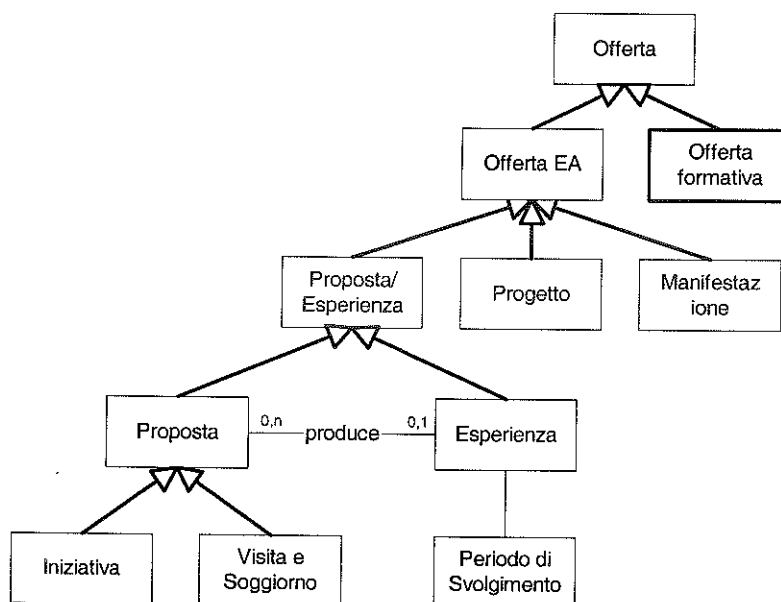


Figura 6

La gerarchia di Figura 6 è stata risolta definendo una relazione *OffertaEA*, che rappresenta il mondo Andrea e una relazione *OffertaFormativa* (*Corsi*) che comprende il mondo Anfora. Non si è ritenuto utile definire una relazione *Offerta* con le proprietà comuni delle due relazioni, poiché, analizzando il traffico delle transazioni, non esistono casi in cui le due relazioni vengono coinvolte contemporaneamente, escluso per interrogazioni relative all'associazione di queste con l'Ente o relative a tutte le offerte a prescindere dalla loro tipologia (in questo caso l'operazione viene risolta tramite una Union tra gli attributi comuni). Per ciascuna delle classi è stata definita una relazione ad eccezione di *Proposta/Esperienza* in quanto il suo unico attributo ha la funzione di distinguere la *Proposta* dall'*Esperienza*. Per lo stesso motivo non viene definita una relazione *Proposta* in quanto la sua unica funzione sarebbe quella di discriminare tra *Iniziativa* e *Visita/Soggiorno* (conterrebbe solo gli ID delle differenti offerte e quindi costringerebbe comunque a fare un join con la relazione opportuna per ottenere informazioni specifiche della Proposta, con questa soluzione si elimina la necessità di un join tra tre relazioni). Poiché la classe *Esperienza* è caratterizzata da un periodo di svolgimento che può essere composto da più anni non consecutivi e, tale periodo, deve potere essere soggetto ad interrogazioni è stato ritenuto opportuno introdurre una nuova relazione *PeriodoDiSvolgimento* che riporti per ciascuna esperienza i periodi nei quali si è svolta. Lo schema diventa allora quello di Figura 7.

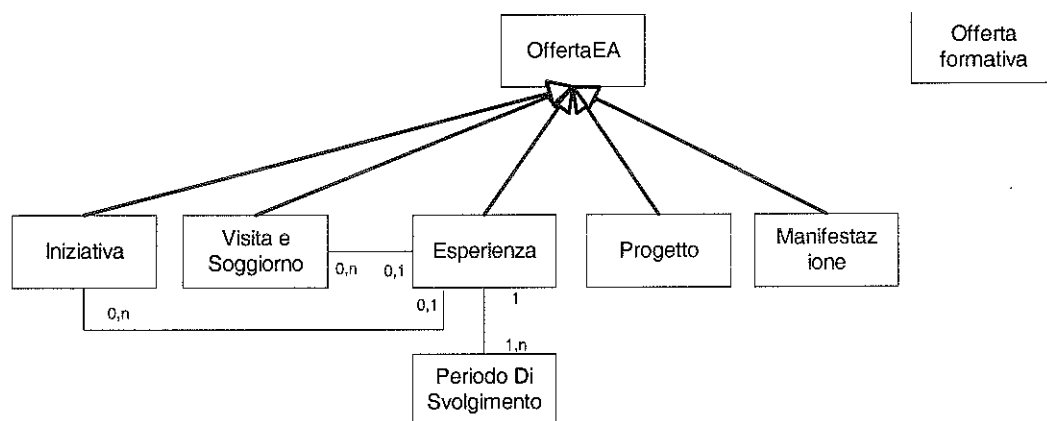


Figura 7

Il nome della relazione associata alla classe *OffertaEA* è *Offerta*, mentre per la classe *Offerta Formativa* vengono definite le relazioni *Corsixxx* (xxx = Universitari, Istruzione, ecc.),

Relazione Offerta

```
CREATE TABLE Offerta (  
    IdOfferta char (8) NOT NULL ,  
    Titolo varchar (255) NOT NULL ,  
    CodTipologiaOfferta smallint NOT NULL ,  
    Descrizione text NULL ,  
    DataInserimento datetime NOT NULL,  
    DataUltimaModifica datetime NOT NULL,  
    UtentePrimoInserimento varchar (20) NOT NULL ,  
    UtenteProponente varchar (20) NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Offerta PRIMARY KEY CLUSTERED  
    (  
        IdOfferta  
    )  
)
```

Questa relazione contiene tutte le offerte. La tipologia di offerta viene discriminata dall'attributo *CodTipologiaOfferta* che può assumere uno dei seguenti valori:

CodTipologiaOfferta	
01	Esperienza
02	Manifestazione
03	Progetto
04	Iniziativa
05	Visita o Soggiorno

Si è preferito non definire una relazione per la classe Proposta/Esperienza in quanto dotata di un unico attributo per la distinzione tra Proposta ed Esperienza. Le proprietà *DataUltimaModifica* e *DataInserimento* sono utili per la gestione del sistema (esempio per poter segnalare le modifiche o le novità).

Relazione Iniziativa

```
CREATE TABLE Iniziativa (  
    IdIniziativa char (8) NOT NULL ,  
    CodTipologiaIniziativa smallint NOT NULL ,  
    Referente varchar (255) NULL ,  
    UrlPaginaWeb varchar (255) NULL ,  
    CONSTRAINT PK_Iniziativa PRIMARY KEY CLUSTERED (  
        IdIniziativa  
    ), CONSTRAINT FK_Iniziativa FOREIGN KEY  
    (  
        IdIniziativa  
    ) REFERENCES dbo.Offerta (  
        IdOfferta  
    )  
)
```

Questa relazione contiene tutte le Iniziative. . Il campo IdIniziativa è chiave primaria e contiene lo stesso valore del campo IdOfferta della generica offerta. La tipologia di iniziativa viene discriminata dall'attributo *CodTipologiaIniziativa* che può assumere uno dei seguenti valori:

CodTipologiaIniziativa	
01	Città dei bambini
02	Percorso didattico
03	Progetto educativo
04	Promozione
05	Ricerca educativa
06	Aggiornamento

Nota: Aggiornamento si riferisce alle attività di aggiornamento per insegnanti.

Relazione Visita/Soggiorno

```
CREATE TABLE VisitaSoggiorno (
    IdVisita char (8) NOT NULL ,
    CodTipologia smallint NOT NULL ,
    Durata varchar (255) NULL ,
    Contatto varchar (255) NULL ,
    CodIscrizione varchar (255) NULL ,
    Accompagnatore char (1) NULL ,
    MaxPersone smallint NULL ,
    IdLocazione char (8) NULL ,
    CONSTRAINT PK_VisitaSoggiorno PRIMARY KEY CLUSTERED
    (
        IdVisita
    ) , CONSTRAINT FK_VisitaSoggiorno FOREIGN KEY
    (
        IdVisita
    ) REFERENCES dbo.Offerta (
        IdOfferta
    )
)
```

Questa relazione contiene tutte le Visite/Soggiorno.. Il campo IdVisita è chiave primaria e contiene lo stesso valore del campo IdOfferta della generica offerta La tipologia viene discriminata dall'attributo *CodTipologia* che può assumere uno dei seguenti valori:

CodTipologia	
01	Campo scuola
02	Escursione
03	Soggiorno
04	Soggiorno estivo
05	Visita guidata

Il tipo di iscrizione viene discriminata dall'attributo *CodIscrizione* che può assumere uno dei seguenti valori:

CodIscrizione	
01	Individuale
02	Di gruppo

L'attributo accompagnatore assume un valore di verità (si/no)

La proprietà *IdLocazione* rappresenta l'associazione tra la classe *VisitaSoggiorno* e la classe *Locazione*. dello schema concettuale ed assume il valore di *IdVisita* oppure di *IdEnte* se la locazione corrisponde a quella dell'ente.

Relazione Esperienza

```
CREATE TABLE Esperienza (
    IdEsperienza char (8) NOT NULL ,
    IdProposta char (8) NOT NULL ,
    CodTipologiaEsperienza smallint NOT NULL ,
    CodTipologiaProposta smallint NOT NULL ,
    Referente varchar (255) NULL ,
    Iperscheda varchar (255) NULL ,
    UrlIperscheda varchar (255) NULL ,
    CONSTRAINT PK_Esperienza PRIMARY KEY CLUSTERED
    (
        IdEsperienza
    ), CONSTRAINT FK_Esperienza FOREIGN KEY
    (
        IdEsperienza
    ) REFERENCES dbo.Offerta (
        IdOfferta
    )
)
```

Questa relazione contiene tutte le Esperienze. Il campo *IdEsperienza* è chiave primaria e contiene lo stesso valore del campo *IdOfferta* della generica offerta. La tipologia di esperienza viene discriminata dall'attributo *CodTipologiaEsperienza* che può assumere uno dei seguenti valori:

CodTipologiaEsperienza	
01	Campo scuola
02	Città dei bambini
03	Escursione
04	Percorso didattico
05	Promozione
06	Ricerca educativa
07	Soggiorno
08	Soggiorno estivo
09	Progetto educativo
10	Visita guidata

L' *IdProposta* riferisce un'offerta che è nella relazione *Iniziative* o *VisiteSoggiorni*. Poiché il periodo di svolgimento può essere composto da più anni non consecutivi

all'interno dei quali è necessario fare delle ricerche una soluzione può essere inserire gli anni in una relazione a parte (PeriodoDiSvolgimento).

Il *CodTipologiaProposta* serve a discriminare la tipologia della proposta, nel caso in cui un'esperienza è correlata ad una proposta e può assumere uno dei seguenti valori.

CodTipologiaProposta	
04	Iniziativa
05	Visita o Soggiorno

Relazione PeriodoDiSvolgimento

```
CREATE TABLE PeriodoDiSvolgimento (  
    IdOfferta char (8) NOT NULL ,  
    AnnoIniziale smallint NOT NULL ,  
    AnnoFinale smallint NULL ,  
    CONSTRAINT PK_PeriodoDiSvolgimento PRIMARY KEY CLUSTERED  
    (  
        IdEsperienza,  
        AnnoIniziale  
    ), CONSTRAINT FK_PeriodoDiSvolgimento FOREIGN KEY  
    (  
        IdOfferta  
    ) REFERENCES dbo.Offerta (  
        IdOfferta  
    )  
)
```

- Se l'esperienza si svolge in un solo anno questo è riportato nel campo AnnoIniziale
- Se l'esperienza si svolge in un periodo di anni consecutivi l'anno iniziale e quello finale sono riportati rispettivamente nei campi AnnoIniziale ed AnnoFinale
- Se l'esperienza si svolge in più anni non consecutivi ciascun anno viene riportato nell'attributo AnnoIniziale
- Si possono avere combinazioni delle precedenti possibilità.

Relazione Progetti

```
CREATE TABLE Progetti (  
    IdProgetto char (8) NOT NULL ,  
    CodAzione smallint not null,  
    Responsabile varchar (255) NOT NULL ,  
    Partecipanti text NULL ,  
    DataAttivazione char (16) NOT NULL ,  
    DurataInMesi smallint NOT NULL ,  
    UrlPaginaWeb varchar (255) NULL ,  
    CONSTRAINT PK_Progetto PRIMARY KEY CLUSTERED (  
        IdProgetto  
    ), CONSTRAINT FK_Progetto FOREIGN KEY (  
        IdProgetto  
    ) REFERENCES dbo.Offerta (  
        IdOfferta  
    )  
)
```

CodAzione	
01	Agenda 21
02	Domeniche Ecologiche
03	Educazione Ambientale
04	Informazione Ambientale
05	Life Ambiente
06	Premio Città dei bambini e delle bambine
07	Premio Città Sostenibili

Relazione Manifestazione

```

CREATE TABLE Manifestazione (
    IdManifestazione char (8) NOT NULL ,
    CodTipologiaManifestazione smallint NOT NULL ,
    Contatto varchar (255) NULL ,
    DataInizio datetime NOT NULL ,
    DataFine datetime NOT NULL ,
    IdLocazione char (8) NULL ,
    UrlPaginaWeb varchar (255) NULL ,
    UrlFileAllegato varchar (255) NULL ,
    PathFileAllegato varchar (255) NULL ,
    CONSTRAINT PK_Manifestazione PRIMARY KEY CLUSTERED
    (
        IdManifestazione
    ) , CONSTRAINT FK_Manifestazione FOREIGN KEY
    (
        IdManifestazione
    ) REFERENCES dbo.Offerta (
        IdOfferta
    )
)

```

Questa relazione contiene tutte le Manifestazioni. . Il campo *IdManifestazione* è chiave primaria e contiene lo stesso valore del campo *IdOfferta* della generica offerta. La tipologia di manifestazione viene discriminata dall'attributo *CodTipologiaManifestazione* che può assumere uno dei seguenti valori:

CodTipologiaManifestazione	
01	Campagna
02	Concorso
03	Conferenza/Convegno
04	Seminario
05	Mostra
06	Settimana ambientale

La proprietà *IdLocazione* rappresenta l'associazione tra la classe *Manifestazione* e la classe *Locazione* dello schema concettuale. *IdLocazione* non è obbligatorio per questa relazione. Per il valore 06 (Settimana ambientale) occorre indagare meglio.

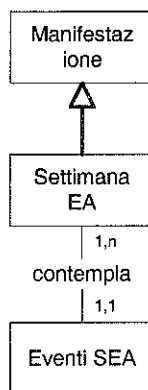


Figura 8

La gerarchia di Figura 8 è stata risolta definendo la relazione *EventiSea*, infatti la classe *Settimana EA* era stata introdotta a livello concettuale per via dell'associazione con *EventiSea*.

Relazione EventiSEA

```

CREATE TABLE EventiSEA (
  IdSEA char (8) NOT NULL ,
  IdEvento char (8) NOT NULL ,
  IdEnte char (8) NOT NULL ,
  Titolo varchar (255) NOT NULL ,
  Contatto varchar (255) NOT NULL ,
  Descrizione text NULL ,
  IdLocazione char (8) NULL ,
  CONSTRAINT PK_Sea PRIMARY KEY CLUSTERED
  (
    IdSea, IdEvento,
  ), CONSTRAINT FK_Sea FOREIGN KEY
  (
    IdSea
  ) REFERENCES dbo.Manifestazione (
    IdManifestazione
  )
)

```

IdLocazione assume lo stesso valore di *IdEnte* o di *IdEvento* a secondo della sede di svolgimento. Occorre indagare meglio su questa relazione alla luce dell'esperienza della III settimana Ea. *IdSea* identifica

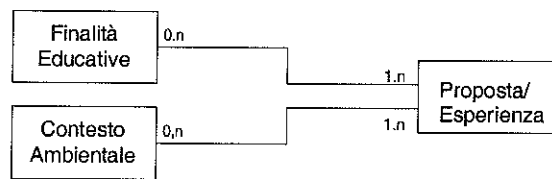


Figura 9

Le classi *ContestoAmbientale* e *FinalitàEducative* e le associazioni con la classe *Proposta/Esperienza* vengono realizzate tramite le relazioni *ContestoAmbientale* e *FinalitàEducative*.

Relazione ContestoAmbientale

```

CREATE TABLE ContestoAmbientale (
  IdOffertaMateriale char (8) NOT NULL ,
  CodContesto smallint NOT NULL ,
  CONSTRAINT PK_ContestoAmbientale PRIMARY KEY CLUSTERED (
    IdOffertaMateriale,
    CodContesto
  )
)
  
```

CodContesto	
01	Ambiente antropizzato
02	Acque dolci
03	Ambiente litoraneo
04	Ambiente marino
05	Ambiente roccioso
06	Ambiente urbano
07	Bosco
08	Prato
09	Zone umide

Relazione FinalitaEducativa

```

CREATE TABLE FinalitaEducativa (
  IdOffertaMateriale char (8) NOT NULL ,
  CodFinalita smallint NOT NULL ,
  CONSTRAINT PK_FinalitaEducativa PRIMARY KEY CLUSTERED (
    IdOffertaMateriale,
    CodFinalita
  )
)
  
```

CodFinalita	
01	Conoscere i problemi dell'ambiente
02	Conoscere l'ambiente
03	Fare esperienza nell'ambiente
04	Intervenire sull'ambiente
05	Riflettere sull'educazione ambientale

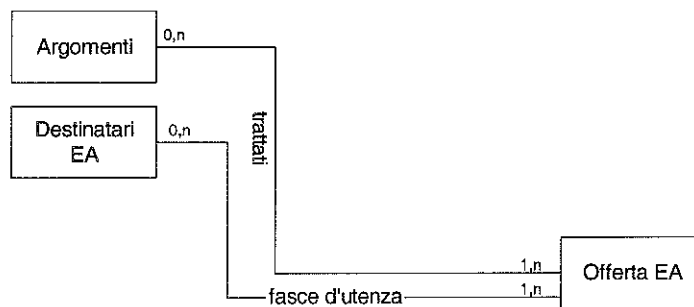


Figura 10

Le classi e le associazioni di Figura 10 sono realizzate con le seguenti relazioni.

Relazione DestinatariEA

```

CREATE TABLE DestinatariEA (
    IdOffertaMaterialeServizio char (8) NOT NULL ,
    CodDestinatari smallint NOT NULL ,
    CONSTRAINT PK_Destinatariea PRIMARY KEY CLUSTERED
    (
        IdOffertaMaterialeServizio,
        CodDestinatari,
    )
)
  
```

Contiene i destinatari delle istanze della classe Offerta EA, Materiali e Servizi. L'attributo *IdOffertaEAMaterialeServizio* permette di correlare gli elementi della relazione con gli elementi di una delle tre relazioni. I valori degli identificatori sono disgiunti.

CodDestinatari	
01	Bambini
02	Giovani
03	Adulti
04	Anziani
05	Tutti

Relazione Argomenti

```

CREATE TABLE Argomenti (
    IdOffertaMateriale char (8) NOT NULL ,
    CodArgomento smallint NOT NULL ,
    IdArgomento smallint NOT NULL ,
    CONSTRAINT PK_Argomenti PRIMARY KEY CLUSTERED (
        IdOffertaMateriale,
        CodArgomento
    )
)
  
```

La relazione descrive la classe Argomenti associata ad OffertaEA e a Materiali. La definizione degli argomenti è quella del Thesaurus di ANDREA. *IdArgomento* è l'identificatore univoco presente in Andrea, *CodArgomento* è l'identificatore univoco creato nella costruzione dell'albero del thesaurus.

```
CREATE TABLE UrlsArgomenti (  
    IdArgomento smallint NOT NULL ,  
    Url varchar (255),  
    CONSTRAINT PK_UrlArgomenti PRIMARY KEY CLUSTERED  
    (  
        IdArgomento  
    )  
)
```

Per Alcuni argomenti esiste una descrizione associata tramite url.

2.1.4 Trasformazione della classe Materiali

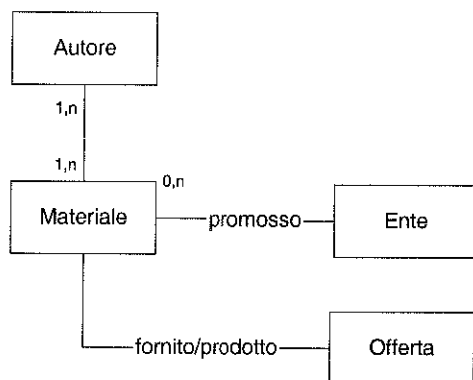


Figura 11

Le classi e le associazioni di Figura 11 sono state implementate tramite le seguenti relazioni.

Relazione Materiali

```
CREATE TABLE Materiali (  
    IdMateriale char (8) NOT NULL ,  
    Titolo varchar (255) NOT NULL ,  
    CodTipologiaMateriale smallint NOT NULL ,  
    Abstract text NULL ,  
    CodLingua smallint NOT NULL ,  
    Editore varchar (255) NULL ,  
    AnnoEdizione smallint NULL ,  
    LuogoEdizione varchar (255) NULL ,  
    AltriRiferimentiBibliografici varchar (255) NULL ,  
    CollocatoInInfea char(1) NOT NULL ,  
    UrlFileAllegato varchar (255) NULL ,  
    PathFileAllegato varchar (255) NULL ,  
    DataPrimoInserimento datetime NOT NULL ,
```

```

DataUltimaModifica datetime NOT NULL ,
UtentePrimoInserimento varchar (20) NOT NULL ,
UtenteProponente varchar (20) NOT NULL ,
UtenteApprovante varchar (20) NOT NULL ,
NoteAmministratore text NULL ,
CONSTRAINT PK_Materiali PRIMARY KEY CLUSTERED (
    IdMateriale
)
)

```

Questa relazione contiene tutti i materiali, che vengono identificati univocamente tramite l'attributo *IdMateriale* che è costituito da una sequenza di 8 caratteri, come per tutti gli Id. Nella codifica dei valori per tale Id si usano i primi 3 caratteri per riferire l'oggetto (es. AUD per Audiovisivo, GIO per Gioco, etc) e gli altri 5 come numero progressivo.

L'attributo *CodTipologiaMateriale* può assumere uno dei seguenti valori:

CodTipologiaMateriale	
01	Audiovisivo
02	Gioco
03	Ipermedia
04	Pannello per mostra
05	Pubblicazione

L'attributo *CodLingua* può assumere uno dei seguenti valori:

CodLingua	
01	Italiano
02	Inglese
03	Francese
04	Spagnolo
05	Tedesco
06

Relazione Autori

```

CREATE TABLE Autori (
    IdMateriale char (8) NOT NULL ,
    Autore char (8) NOT NULL ,
)

```

Questa relazione contiene tutti i dati relativi agli autori dei materiali. Per semplicità di realizzazione non è stata normalizzata la doppia associazione multipla, per cui la relazione può contenere valori duplicati e per questo motivo non viene definita nessuna chiave primaria.

Relazione MaterialiEnti

```
CREATE TABLE MaterialiEnti (  
    IdMateriale char (8) NOT NULL ,  
    IdEnte char (8) NOT NULL ,  
    CONSTRAINT PK_MaterialiEnti PRIMARY KEY CLUSTERED  
    (  
        IdEnte,  
        IdMateriale  
    )  
)
```

Questa relazione rappresenta l'associazione "promosso" dello schema concettuale.

Relazione OffertaMateriali

```
CREATE TABLE OffertaMateriali (  
    IdMateriale char (8) NOT NULL ,  
    IdOfferta char (8) NOT NULL ,  
    CONSTRAINT PK_OffertaMateriali PRIMARY KEY CLUSTERED  
    (  
        IdMateriale,  
        IdOfferta  
    )  
)
```

Questa relazione rappresenta l'associazione "fornito/prodotto" dello schema concettuale. L'interpretazione dell'associazione avviene tramite la relazione con cui si fa join (*fornito* nel caso di join con un'Offerta di tipo Proposta e *prodotto* nel caso di altre tipologie di Offerta). Con questa modellazione è possibile associare dei materiali a qualsiasi Offerta, per cui occorre prevedere a livello di interfaccia eventuali vincoli (se esistono per esempio delle Offerte che non producono materiali).

2.1.5 Trasformazione della classe Locazione

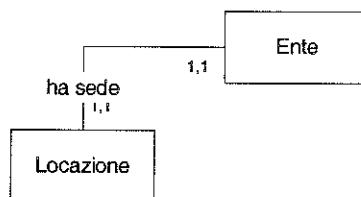


Figura 4b. Associazione Ente/Locazione

L'associazione "ha sede" tra la classe *Ente* e la classe *Locazione* dello schema concettuale viene realizzata tramite la corrispondenza *IdEnte* ed *IdLocazione* della relazione *Locazione*.

Questa relazione si tiene separata sebbene potrebbe confluire, per la classe *Ente*, nella relazione *Enti* essendo l'associazione che ad essa collegata del tipo (1:1) in entrambi i versi. Per uniformità di trattamento e di gestione con le classe *Servizi*, *Materiali*, ecc., poiché in questo caso l'associazione è opzionale, si preferisce la seguente scelta.

Relazione *Locazione*

```
CREATE TABLE Locazione (  
    IdLocazione char (8) NOT NULL ,  
    Stato varchar (255) NULL ,  
    SiglaProvincia char(2) NOT NULL ,  
    Citta varchar (255) NOT NULL ,  
    Cap integer NOT NULL,  
    Indirizzo varchar (255) NOT NULL ,  
    Collegamenti text NULL ,  
    Strutture text NULL,  
    BarriereArchitettoniche Char (1) NOT NULL ,  
    DataPrimoInserimento datetime NOT NULL ,  
    DataUltimaModifica datetime NOT NULL ,  
    UtentePrimoInserimento varchar (20) NOT NULL ,  
    UtenteProponente varchar (20) NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_ Locazione PRIMARY KEY CLUSTERED  
    (  
        IdLocazione  
    )  
)
```

Questa relazione contiene tutte le specifiche delle locazioni, per cui l'*IdLocazione* corrisponde all'*IdEnte* se la locazione è relativa all'*Ente*, l'*IdServizio* se relativa al servizio, l'*IdMateriale* se relativo al *Materiale*, etc. Tutti i dati di questa tabella (escluso la chiave) possono essere duplicati per gli oggetti del DB che hanno l'identica specifica locazione. Questa scelta semplifica la sua gestione, infatti se supportassimo l'associazione multipla, dovremmo complicare notevolmente la gestione dell'aggiornamento (prevedendo ad esempio che l'aggiornamento dell'indirizzo riguarda solo alcuni degli oggetti che puntavano al vecchio indirizzo, etc).

Relazione ContestoTerritoriale

```
CREATE TABLE ContestoTerritoriale (  
    CodAreaGeografica smallint NOT NULL ,  
    CodRegione smallint NOT NULL ,  
    CONSTRAINT PK_ContestoTerritoriale PRIMARY KEY CLUSTERED (  
        CodAreaGeografica, CodRegione  
    )  
)
```

Questa relazione contiene i contesti territoriali, così come definita da *CodAreaGeografica*, tramite i quali selezionare le ennuple nella relazione *Locazione*. Si è preferito tenere separata questa relazione per fornire la possibilità di definire ulteriori contesti per cui raggruppare le ennuple della relazione *Locazione*.

La proprietà *CodAreaGeografica* assume i seguenti valori:

CodAreaGeografica	
01	Italia centrale
02	Italia insulare
03	Italia meridionale
04	Italia nord occidentale
05	Italia nord orientale

Relazione Regioni

```
CREATE TABLE Regioni (  
    CodRegione smallint NOT NULL ,  
    Regione varchar (30) NOT NULL ,  
    CONSTRAINT PK_Regioni PRIMARY KEY CLUSTERED  
    (  
        CodRegione  
    )  
)
```

Relazione Provincie

```
CREATE TABLE Provincie (  
    SiglaProvincia char(2) NOT NULL ,  
    CodRegione smallint NOT NULL ,  
    Provincia varchar (30) NOT NULL ,  
    CONSTRAINT PK_Provincie PRIMARY KEY CLUSTERED (  
        CodProvincia  
    )  
)
```

2.1.6 Trasformazione della classe Agenda

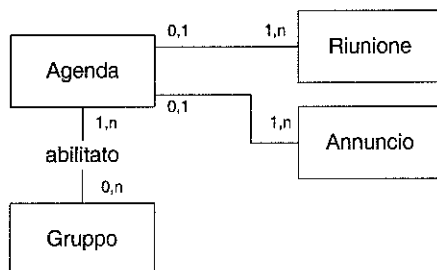


Figura 12

Le classi e le associazioni di Figura 12 sono state implementate tramite le seguenti relazioni.

Relazione Agenda

```
CREATE TABLE Agenda (  
    IdEvento char (8) NOT NULL ,  
    CodTipologiaEvento smallint NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    DataInserimentoInAgenda datetime NOT NULL ,  
    DataScadenza datetime NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Agenda PRIMARY KEY CLUSTERED  
    (  
        IdEvento  
    )  
)
```

Questa relazione contiene gli eventi in Agenda. I vari eventi sono identificati da *IdEvento* e fanno riferimento tramite la proprietà *CodTipologiaEvento* alla Manifestazioni o Segnalazioni.. La relazione viene popolata dall'Activity Manager dell'agenda selezionando gli eventi significativi per la segnalazione. La data di scadenza identifica il periodo di validità in Agenda che in generale può essere diverso dalle date di validità dell'evento. La proprietà *CodTipologiaEvento* assume i seguenti valori.

CodTipologiaEvento	
01	Manifestazione
02	Segnalazione

Relazione Segnalazioni

```
CREATE TABLE Segnalazioni (  
  IdSegnalazione char (8) NOT NULL ,  
  Titolo varchar (255) NOT NULL ,  
  Descrizione text NULL ,  
  UrlPaginaWeb varchar (255) NULL ,  
  UrlFileAllegato varchar (255) NULL ,  
  PathFileAllegato varchar (255) NULL ,  
  DataInizio datetime NOT NULL ,  
  DataFine datetime NULL ,  
  DataPrimoInserimento datetime NOT NULL ,  
  DataUltimaModifica datetime NOT NULL ,  
  UtentePrimoInserimento varchar (20) NOT NULL ,  
  UtenteProponente varchar (20) NOT NULL ,  
  UtenteApprovante varchar (20) NOT NULL ,  
  CONSTRAINT PK_Segnalazioni PRIMARY KEY CLUSTERED  
  (  
    IdSegnalazione  
  )  
)
```

Questa relazione contiene gli eventi da segnalare in agenda che non rientrano nelle categorie previste dalla classe Offerta (esempio: “il 28 cm il ministro inaugura il nuovo LT di Piazza Armerina”). L’eventuale Locazione dell’evento compare in forma descrittiva nella Descrizione.

2.2. Trasformazione della classe Utenti e gruppi

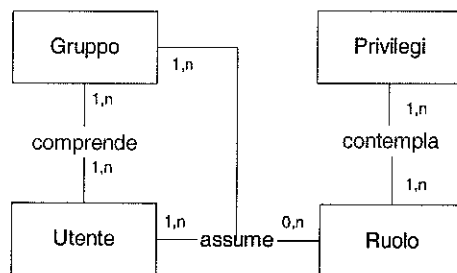


Figura 13

Le classi e le associazioni di Figura 13 sono implementate tramite le seguenti relazioni.

Relazione Utenti

```
CREATE TABLE Utenti (  
    IdUtente char (12) NOT NULL ,  
    IdGruppoPredefinito char (12) NOT NULL ,  
    Utente varchar (20) NOT NULL ,  
    Nome varchar (50) NOT NULL ,  
    Cognome varchar (50) NOT NULL ,  
    Password varchar (20) NOT NULL ,  
    Email varchar (255) NOT NULL ,  
    CodStatoUtente smallint NOT NULL,  
    IdEnte char(8) NOT NULL,  
    Descrizione varchar (255) NULL ,  
    DataPrimoInserimento datetime NOT NULL ,  
    DataUltimaModifica datetime NOT NULL ,  
    IdUtenteApprovante char (12) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Utenti PRIMARY KEY CLUSTERED  
    (  
        IdUtente  
    )  
)
```

CodStatoUtente	
01	Abilitato
02	Disabilitato
03 (usi futuri)

Relazione Gruppi

```
CREATE TABLE Gruppi (  
    IdGruppo char (12) NOT NULL ,  
    NomeGruppo varchar (20) NOT NULL ,  
    SiglaGruppo char (2) NOT NULL ,  
    Descrizione varchar (255) NULL ,  
    DataPrimoInserimento datetime NOT NULL ,  
    DataUltimaModifica datetime NOT NULL ,  
    IdUtenteApprovante char (12) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_Gruppi PRIMARY KEY CLUSTERED  
    (  
        IdGruppo  
    )  
)
```

La *SiglaGruppo* serve per eventuali realizzazioni distribuiti del sito per evitare la generazione di chiavi duplicati, il valore viene usato nella generazione degli

identificatori dei vari oggetti del database. Ad ogni gruppo è associato almeno un utente col ruolo *Redazione*, a cui arrivano le notifiche delle richieste di aggiornamento.

Relazione UtentiGruppi

```
CREATE TABLE UtentiGruppi (  
  IdUtente char (12) NOT NULL ,  
  IdGruppo char (12) NOT NULL ,  
  CodRuolo smallint NOT NULL,  
  DataPrimoInserimento datetime NOT NULL ,  
  DataUltimaModifica datetime NOT NULL ,  
  IdUtenteApprovante char (12) NOT NULL ,  
  NoteAmministratore text NULL,  
  CONSTRAINT PK_UtentiGruppi PRIMARY KEY CLUSTERED (  
    NomeGruppo,  
    Utente,  
    CodRuolo  
  )  
)
```

Un utente assume un solo ruolo, non necessariamente il solito, per ogni gruppo di cui fa parte.

CodRuolo	
01	Ospite
02	Labnet
03	Redazione
04	GestioneFinanziaria
05	Amministratore del sito
06	Dirigente
07	Amministratore di sistema
08 (usi futuri)

Il ruolo identifica un insieme di privilegi.

Relazione PrivilegiRuolo

```
CREATE TABLE PrivilegiRuolo (  
  CodRuolo smallint NOT NULL,  
  CodOggetto smallint NOT NULL ,  
  CodOperazione smallint NOT NULL ,  
  DataPrimoInserimento datetime NOT NULL ,  
  DataUltimaModifica datetime NOT NULL ,  
  IdUtenteApprovante char (12) NOT NULL ,  
  NoteAmministratore text NULL ,  
  CONSTRAINT PK_Autorizzazioni PRIMARY KEY CLUSTERED (  
    CodRuolo,  
    CodOggetto,  
    CodOperazione  
  )  
)
```

CodOggetto corrisponde ad un oggetto del database. Per ogni ruolo è definito un template che specifica tutti gli oggetti e tutte le operazioni ammissibili su di esso. Nel caso si volessero imporre restrizioni o maggiori privilegi ai ruoli esistenti si definisce un nuovo ruolo con un nuovo template.

CodOggetto	
01	Agenda
02	Enti
03	Iniziative
05	Manifestazioni
06	Visite e soggiorni
07	Esperienze
08	Materiali
09	Offerta educativa
10	Progetti
11	Corsi di formazione
12	Corsi universitari
13	Corsi Master
14	Corsi istruzione secondaria
15	Corsi IFTS
16	Offerta formativa
17	Gestione finanziaria
18	Utenti e ruoli
100	Tutti

La registrazione di un utente è permessa a tutti i ruoli, la gestione di *Utenti e ruoli* è consentita solo al ruolo di redazione (o ad un ruolo più elevato).

Offerta educativa contempla tutti gli oggetti che la compongono compreso Enti ed Agenda. *Offerta formativa* comprende tutti i corsi oltre ad Enti ed Agenda.

CodOperazione	
01	Lettura
02	Proponi Inserimento
03	Proponi modifica
04	Proponi cancellazione
05	Inserimento
06	Modifica
07	Cancellazione

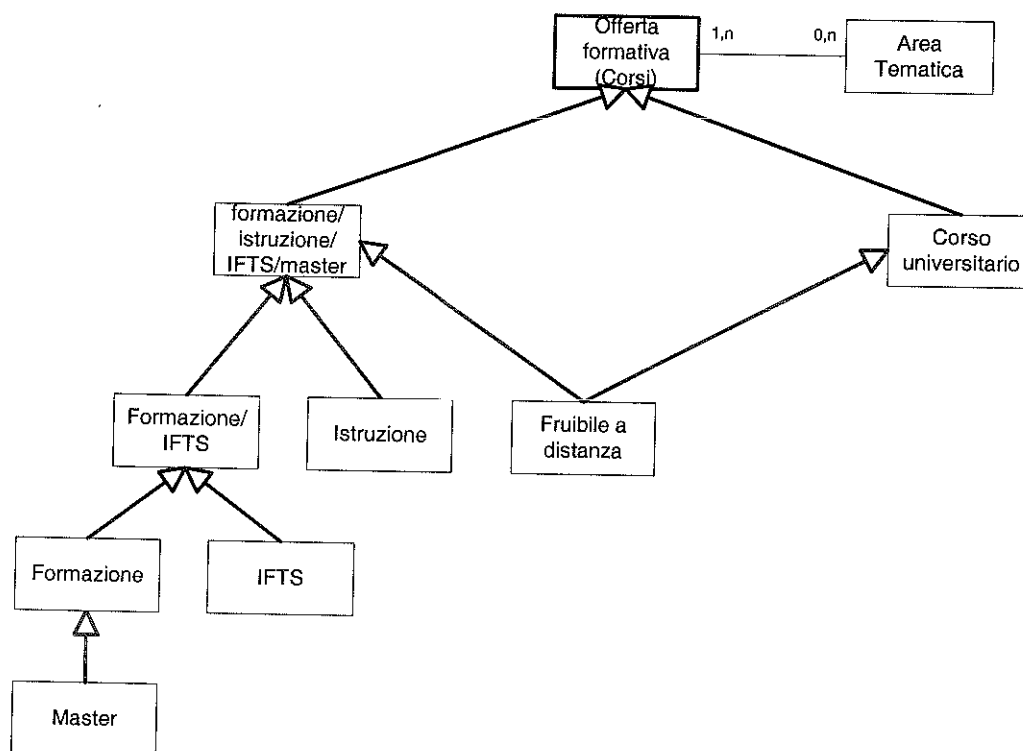
Il numero d'ordine di un'operazione implica tutte le operazioni precedenti.

2.3 Trasformazione della classe Offerta Formativa

Il tentativo di trattare l'*Offerta Formativa* come tutte le altre offerte, si scontra con le differenti definizioni assunte da Andrea ed Anfora, per le classi *Argomenti*, e *Destinatari*. Tale convergenza non si è verificata in quanto la definizione di un approccio comune implicava un notevole impegno di studio, in cui venissero coinvolti esperti nella definizione di thesauri, non affrontabile con le risorse e gli impegni previsti per questo progetto.

La relazione *OffertaEnti* potrebbe essere clonata nella relazione *CorsiEnti* (qualora se ne presentasse la necessità per motivi di efficienza).

La risoluzione della gerarchia IS-A è avvenuta per partizionamento, dopo aver osservato che il traffico delle richieste che coinvolgono tutti i corsi a prescindere dalla loro natura coinvolge solo filtri su *ContestoTerritoriale*, *Temi* ed *EntiCoinvolti* e che le proprietà comuni a tutti i corsi si riducevano al solo *Titolo*. Questa soluzione naturalmente complica le richieste che riguardano tutti i corsi a prescindere dalla loro natura (occorre fare *Union* tra tutte le tipologie di corsi), ma semplifica tutte le altre su corsi specifici in quanto non obbliga a fare *join* per rilevare il titolo del corso.



Per cui la gerarchia in figura è stata risolta definendo le relazioni:

CorsoDiFormazione,
CorsiIstruzioneSecondaria,
CorsiMaster,
CorsiIFTS,
CorsiUniversitari
CorsiFad

Relazione AreeTematiche

```
CREATE TABLE AreeTematiche (  
  IdCorso char (8) NOT NULL ,  
  CodTema smallint NOT NULL ,  
  CONSTRAINT PK_Temi PRIMARY KEY CLUSTERED  
  (  
    IdCorso,  
    CodTema  
  )  
)
```

Questa relazione rappresenta l'associazione multipla tra le classi *OffertaFormativa* e *AreaTematica*.

<i>CodTema</i>	
01	acqua
02	agricoltura
03	aria
04	beni culturali e ambientali
05	conservazione della natura
06	ecologia
07	informazione ed educazione ambientale
08	energia
09	gestione e pianificazione
10	igiene e sanità
11	impatto ambientale
12	normativa ambientale
13	rifiuti
14	rumore
15	sicurezza del lavoro e dell'ambiente
16	qualità e certificazione
17	tecniche e tecnologie ambientali
18	territorio
19	turismo ambientale
20	urbanistica
21	verde urbano

2.3.1 Trasformazione della classe Corso Universitario

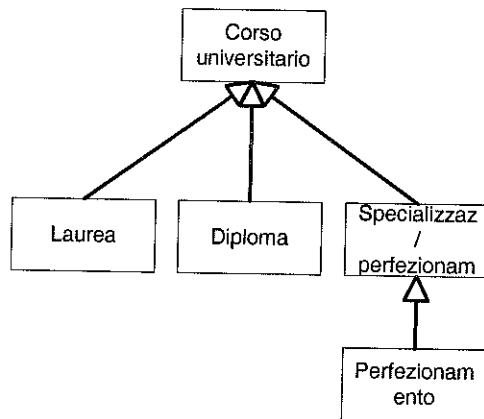


Figura 14

La gerarchia di figura 14 è stata risolta, per comodità di implementazione, definendo un'unica relazione (CorsiUniversitari), che contempla tutti i vari tipi di corsi universitari. Questa scelta deriva dalla constatazione che le classi specializzate di CorsoUniversitario contemplano poche proprietà aggiuntive, per cui sebbene la relazione risultante non è normalizzata, le semplificazioni implementative sono notevoli.

Relazione CorsiUniversitari

```

CREATE TABLE CorsiUniversitari(
    IdCorso char (8) NOT NULL ,
    Facoltà varchar (255) NULL ,
    Dipartimento varchar (255) NULL ,
    CodTipologia smallint NOT NULL ,
    Titolo varchar (255) NOT NULL ,
    IndirizzoDiStudioOrientamento varchar (255) NULL ,
    PeriodoDiSvolgimento varchar (255) NULL ,
    CodDurata smallint NULL ,
    NumeroEsamiOModuli smallint NULL ,
    AnnoAttivazione smallint NULL ,
    NumeroChiuso char(1) NULL ,
    NumeroPosti smallint NULL ,
    TestIngresso char(1) NULL ,
    QuotaIscrizione int NULL ,
    RequisitiDiIngresso text NULL,
    Materie text NULL,
    CollaborazionePartenariato text NULL,
    Referente varchar (255) NULL ,
    DescrizioneContenuto text NULL,
    AltreInformazioni text NULL,
    DataInserimento datetime NOT NULL,
    DataUltimaModifica datetime NOT NULL,
    UtentePrimoInserimento varchar (20) NOT NULL ,
    UtenteProponente varchar (20) NOT NULL ,
    UtenteApprovante varchar (20) NOT NULL ,
    NoteAmministratore text NULL ,
    CONSTRAINT PK_CorsiUniversitari PRIMARY KEY CLUSTERED (
        IdCorso
    )
)
    
```

Nota: la proprietà *IndirizzoDiStudioOrientamento* assume valore solo per Corso di Laurea (Indirizzo di studio) e per Diploma Universitario (Orientamento). Le proprietà *CondizioniDiAccesso*, *QuotaIscrizione* e *Materie* assumono valore solo per i Corsi di specializzazione o di perfezionamento. La proprietà *PeriodoDiSvolgimento* assume valore solo per *Corso di perfezionamento*.

I dati relativi alla *segreteria studenti* ed alla *presidenza/coordinamento del corso* appaiono nei riferimenti dell'Ente.

La proprietà *CodTipologia* assume i seguenti valori:

CodTipologia	
01	Laurea
02	Diploma universitario
03	Scuola diretta a fini speciali
04	Scuola di specializzazione
05	Scuola di perfezionamento.

La proprietà *CodDurata* assume i seguenti valori:

CodDurata	
01	annuale
02	biennale
03	triennale
04	Quadriennale
05	Quinquennale
06

2.3.2 Trasformazione della classe Corso Istruzione

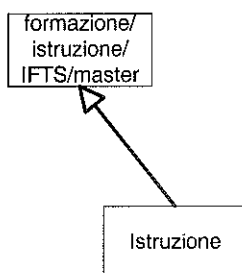


Figura 15

La gerarchia di figura 15 è stata risolta definendo un'unica relazione *CorsiIstruzioneSecondaria* che contiene le proprietà di entrambe le classi. Non è stato ritenuto utile definire due relazioni distinte, poiché è sembrata poco significativa la possibilità di esprimere condizioni di ricerca sulle proprietà comuni alle 4 tipologie di corsi.

Relazione CorsiIstruzioneSecondaria

```
CREATE TABLE CorsiIstruzioneSecondaria (  
  IdCorso char (8) NOT NULL ,  
  CodTipo smallint NULL,  
  Titolo varchar (255) NOT NULL ,  
  DescrizioneContenuto text NULL ,  
  CodDurata smallint NULL ,  
  CodModalitaTemporali smallint NULL ,  
  CollaborazionePartenariato text NULL,  
  AltreInformazioni text NULL,  
  DataInserimento datetime NOT NULL,  
  DataUltimaModifica datetime NOT NULL,  
  UtentePrimoInserimento varchar (20) NOT NULL ,  
  UtenteProponente varchar (20) NOT NULL ,  
  UtenteApprovante varchar (20) NOT NULL ,  
  NoteAmministratore text NULL ,  
  CONSTRAINT PK_CorsiIstruzione PRIMARY KEY CLUSTERED (  
    IdCorso  
  )  
)
```

La proprietà *CodTipo* assume i seguenti valori:

<i>CodTipo</i>	
01	Attestato di qualifica
02	Diploma

La proprietà *CodModalitaTemporali* assume i seguenti valori:

<i>CodModalitaTemporali</i>	
01	Tempo pieno
02	Tempo pieno residenziale
03	Tempo parziale
04	Fine settimana
100	Altra modalità (specificare)

Per altra modalità viene memorizzata soltanto la specifica che assume un valore di codice da 4 in poi, in questo modo è possibile esprimere un filtro su tutti i corsi che assumono valore altra modalità (Cod > 4) che su una specifica altra modalità (es.Cod 18).

Relazione CorsiADistanza

```
CREATE TABLE CorsiADistanza (  
    IdCorso char (8) NOT NULL ,  
    OrganizzazioneInModuli char(1) NOT NULL,  
    CorsiIndividualizzabili char(1) NOT NULL,  
    NumeroSettimane smallint NULL,  
    PercentualeOreADistanza smallint NULL,  
    PercentualeOreInPresenza smallint NULL,  
    FeedbackSuMateriali Char(1) NULL,  
    CONSTRAINT PK_CorsiADistanza PRIMARY KEY CLUSTERED (  
        IdCorso  
    )  
)
```

Questa relazione se popolata è sempre da considerarsi come estensione di una delle altre tipologie di corso.

Relazione ModuliFAD

```
CREATE TABLE ModuliFAD(  
    IdCorso char (8) NOT NULL ,  
    NumProgressivo smallint NOT NULL,  
    Titolo varchar(255) NULL,  
    CONSTRAINT PK_ModuliFAD PRIMARY KEY CLUSTERED (  
        IdCorso , NumProgressivo  
    ),  
    CONSTRAINT FK_ModuliFAD FOREIGN KEY (  
        IdCorso  
    ) REFERENCES dbo.CorsiADistanza (  
        IdCorso  
    )  
)
```

Relazione MaterialiDidatticiFAD

```
CREATE TABLE MaterialiDidatticiFAD (  
    IdCorso char (8) NOT NULL ,  
    CodMateriale smallint NOT NULL,  
    CONSTRAINT PK_MaterialiFad PRIMARY KEY CLUSTERED (  
        IdCorso , CodMateriale  
    ),  
    CONSTRAINT FK_MaterialiFad FOREIGN KEY (  
        IdCorso  
    ) REFERENCES dbo.CorsiADistanza (  
        IdCorso  
    )  
)
```

CodMateriale	
01	<i>Cartaceo</i>
02	<i>Audiovisivo (videocassette, lezioni TV)</i>
03	<i>Cd-Rom</i>
04	<i>Via web</i>

Relazione DotazioniFAD

```

CREATE TABLE DotazioniFAD (
    IdCorso char (8) NOT NULL ,
    CodDotazione smallint NOT NULL,
    Specifica varchar (255) null,
    CONSTRAINT PK_DotazioniFad PRIMARY KEY CLUSTERED (
        IdCorso , CodDotazione
    ),
    CONSTRAINT FK_DotazioniFad FOREIGN KEY (
        IdCorso
    ) REFERENCES dbo.CorsiADistanza (
        IdCorso
    )
)

```

CodDotazione	
01	<i>Videoregistratore</i>
02	<i>Fax</i>
03	<i>Computer</i>
04	<i>Stampante</i>
05	<i>Collegamento Internet</i>
100	<i>altro</i>

Relazione DidatticaFAD

```

CREATE TABLE DidatticaFAD (
    IdCorso char (8) NOT NULL ,
    CodDidatticaFad smallint NOT NULL,
    CONSTRAINT PK_DidatticaFad PRIMARY KEY CLUSTERED (
        IdCorso , CodDidatticaFad
    ),
    CONSTRAINT FK_DidatticaFad FOREIGN KEY (
        IdCorso
    ) REFERENCES dbo.CorsiADistanza (
        IdCorso
    )
)

```

CodDidatticaFad	
01	<i>Test</i>
02	<i>Multimedia</i>
03	<i>Lavagna condivisa</i>
04	<i>Chat</i>
05	<i>Utilizzo banche dati</i>
06	<i>Forum di discussione</i>
07	<i>FAQ</i>
08	<i>Tour in rete</i>
100	<i>altro</i>

Relazione InterazioneCorsiADistanza

```

CREATE TABLE InterazioneCorsiADistanza (
  IdCorso char (8) NOT NULL ,
  CodModalita smallint NOT NULL,
  CodSoggetto smallint NOT NULL,
  CodStrumento smallint NOT NULL,
  SpecificaInPresenza VarChar(255) NULL,
  CONSTRAINT PK_InterazioneFad PRIMARY KEY CLUSTERED (
    IdCorso , CodModalita, CodSoggetto, CodStrumento
  ),
  CONSTRAINT FK_InterazioneFad FOREIGN KEY (
    IdCorso
  ) REFERENCES dbo. CorsiADistanza (
    IdCorso
  )
)

```

La proprietà *SpecificaInPresenza* può assumere valore solo per *CodModalita* = “In presenza”. *CodStrumento* indica gli strumenti di interazione con docenti, tutor e altri partecipanti e assume i valori presenti nella codifica solo per *CodModalita* = “A distanza”, altrimenti assume valore -1. La relazione non è stata normalizzata per semplificazione dell’implementazione.

La proprietà *CodModalita* assume i seguenti valori:

CodModalita	
01	<i>A distanza</i>
02	<i>In presenza</i>

La proprietà *CodSoggetto* assume i seguenti valori:

CodSoggetto	
01	<i>Docenti</i>
02	<i>Tutor</i>
03	<i>Altri partecipanti</i>

La proprietà *CodStrumento* assume i seguenti valori:

CodStrumento	
01	<i>Via telefono</i>
02	<i>Via fax</i>
03	<i>Via e-mail</i>
04	<i>Forum</i>
05	<i>Via chat</i>
06	<i>Videoconferenza</i>
100	<i>altro</i>

2.3.3 Trasformazione della classe Corso Formazione/IFTS/Master

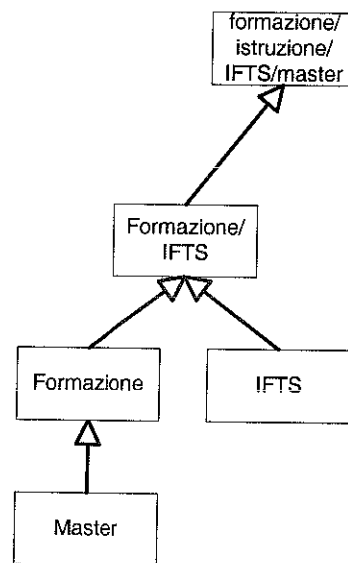


Figura 16

La gerarchia di Figura 16 viene risolta definendo una relazione Per i corsi di Formazione, una per i Master ed una per i corsi IFTS.

Relazione CorsiDiFormazione

```
CREATE TABLE CorsiDiFormazione (
  IdCorso char (8) NOT NULL ,
  Titolo varchar (255) NOT NULL ,
  DescrizioneContenuto text null,
  CodNaturaCorso smallint NULL,
  QuotaIscrizione int NULL,
  MaxAmmessi smallint NULL,
  CostoOraAllievo int NULL,
  CollaborazionePartnenariato text NULL,
  CodTipoGestione smallint NULL,
  CodTipologiaFormativa smallint NULL,
  ModalitàDiAmmissione text NULL,
  CodValutazioneCompetenze smallint NULL,
  CodCertificazioneFinale smallint NULL,
  CodModalitàDidatticaPrevalente smallint NULL,
  CodModalitàTemporal smallint NULL,
  CodDurata smallint NULL ,
  AnnualitàInCorso smallint NULL,
  OreComplessive smallint NULL,
  Attivato char(1) NOT NULL,
  AltreInformazioni text NULL,
  DataInserimento datetime NOT NULL,
  DataUltimaModifica datetime NOT NULL,
  UtentePrimoInserimento varchar (20) NOT NULL ,
  UtenteProponente varchar (20) NOT NULL ,
  UtenteApprovante varchar (20) NOT NULL ,
  NoteAmministratore text NULL ,

  CONSTRAINT PK_CorsiDiFormazione PRIMARY KEY CLUSTERED (
    IdCorso
  )
)
```

La proprietà *CodModalitàTemporal* assume i seguenti valori per tutti i Corsi (escluso Universitari):

<i>CodModalitàTemporal</i>	
01	Tempo pieno
02	Tempo pieno residenziale
03	Tempo parziale
04	Fine settimana
100	Altra modalità (specificare)

<i>CodNaturaCorso</i>	
01	Pubblico
02	Privato

CodTipologiaFormativa	
01	Qualificazione
02	Specializzazione
03	Aggiornamento
04	Perfezionamento
05	Patente di mestiere

CodTipoGestione	
01	Diretta/delegata
02	Convenzionata pubblica
03	Convenzionata privata
04	Gestione di mercato
100	Altro

CodValutazioneCompetenze	
01	Si, in itinere
02	Si, finale
03	no

CodCertificazione finale	
01	Frequenza
02	Qualifica
03	Specializzazione
04	Patente di mestiere/abilitazione professionale
05	Nessuno

CodModalitàDidatticaPrevalente	
01	In aula
02	In alternanza
03	A distanza
100	Altro

Relazione SedeDiSvolgimento

```

CREATE TABLE SedeDiSvolgimento (
  IdCorso char (8) NOT NULL ,
  NumProgressivo smallint NOT NULL,
  IDLocazione Char(8) NULL,
  Cap integer NOT NULL,
  Citta varchar (255) NOT NULL ,
  SiglaProvincia char(2) NOT NULL ,
  Indirizzo varchar (255) NOT NULL ,
  Collegamenti text NULL ,
  Strutture text NULL,
  BarriereArchitettoniche Char (1) NOT NULL ,
  AvvioDaDefinire char(1)NOT NULL,
  Dal datetime NULL,
  Al datetime NULL,
  CONSTRAINT PK_SedeDiSvolgimento PRIMARY KEY CLUSTERED (
    IdCorso , NumProgressivo
  )
)

```

Relazione CondizioniDiAccesso

```
CREATE TABLE CondizioniDiAccesso (  
    IdCorso char (8) NOT NULL ,  
    CodRequisito smallint NOT NULL,  
    SpecificaRequisito varchar(255) NULL,  
    CONSTRAINT PK_CondizioniDiAccesso PRIMARY KEY CLUSTERED (  
        IdCorso, CodRequisito  
    )  
)
```

La proprietà *CodRequisito* assume i seguenti valori:

CodRequisito	
01	Scuola dell'obbligo
02	Licenza media
03	Frequenza scuola secondaria
04	Attestato di qualifica
05	Diploma scuola secondaria
06	Diploma universitario
07	Laurea
08	Conoscenza lingua straniera
100	Altro

La proprietà *SpecificaRequisito* assume un valore non a priori codificabile, dipendente dal valore assunto da *CodRequisito* (es. Laurea in Fisica)

Relazione DestinatariFormazione

```
CREATE TABLE DestinatariFormazione (  
    IdCorso char (8) NOT NULL ,  
    CodDestinatari smallint NOT NULL ,  
    CodSpecifica smallint NOT NULL ,  
    CONSTRAINT PK_DestinatariOFA PRIMARY KEY CLUSTERED (  
        IdCorso,  
        CodDestinatari,  
        CodSpecifica  
    )  
)
```

Contiene i destinatari delle istanze della classe corsi di Formazione. Tutte le specifiche relative alla codifica *Altro* vengono fornite nella proprietà del corso "Altre informazioni rilevanti". Per i corsi Master non sono ammesse le codifiche 02 e 100.

CodDestinatari	
01	Giovani in cerca di I occupazione
02	Soggetti in situazione di svantaggio
03	Donne
04	Disoccupati
05	Occupati
100	Altro

Per le offerte formative può avere un'ulteriore specifica negli attributi *CodDestinatari* e *CodSottocategoriaDestinatari*.

Se *CodDestinatari* = *Giovani in cerca di I occupazione*

CodSpecifica	
01	Drop-out
02	Allievi
03	Giovani con qualifica
04	Giovani con diploma
05	Universitari e neo laureati

Se *CodDestinatari* = *Soggetti in situazione di svantaggio*

CodSpecifica	
01	Portatori di handicap
02	Tossicodipendenti
03	Detenuti ed ex detenuti
04	Immigrati
05	Nomadi
06	Disoccupati adulti in età avanzata

Se *CodDestinatari* = *Disoccupati*

CodSpecifica	
01	Soggetti in riconversione
02	Di lunga durata
03	Mobilità
04	CIG
100	Altro

Se *CodDestinatari* = *Occupati*

CodSpecifica	
01	Apprendisti
02	Contratto formazione lavoro
03	Piccola e media impresa
04	Grande impresa
05	Istruzione e formazione professionale
06	Pubblica amministrazione
100	Altro

Relazione SpecificaDidattica

```
CREATE TABLE SpecificaDidattica(
  IdCorso char (8) NOT NULL ,
  CodModalitaDidattica smallint NOT NULL,
  NumeroOre smallint NULL,
  SpecificaDiAltro varchar (256) NULL,
  CONSTRAINT PK_Didattica PRIMARY KEY CLUSTERED (
    IdCorso,
    CodModalitaDidattica
  )
)
```

La proprietà *CodModalitaDidattica* assume i seguenti valori:

CodModalitaDidattica	
01	Lezioni in aula
02	Esercitazioni
03	Lezioni/esercitazioni a distanza
04	Lavoro in sottogruppo
05	Ricerca di gruppo
06	Discussione dei casi
07	Discussione guidata
08	Simulazione
09	Laboratori
10	Seminari
11	Visite guidate/viaggi di studio
12	Project work (progetti sul campo)
13	Stage
100	altro

Per il trattamento della modalità *altro* si adotta la stessa tecnica prevista per *CodModalitaTemporali*

Relazione Docenza

```
CREATE TABLE Docenza(  
  IdCorso char (8) NOT NULL ,  
  CodTipoDocenza smallint NULL,  
  SpecificaDiAltro varchar (256) NULL,  
  CONSTRAINT PK_Docenza PRIMARY KEY CLUSTERED (  
    IdCorso, CodTipoDocenza  
  )  
)
```

La proprietà *CodTipoDocenza* assume i seguenti valori:

CodTipoDocenza	
01	universitaria
02	aziendale
03	consulenza
04	testimonianza
05	tutoring
06	animatori
07	funzionari di P.A.
08	docenti interni
09	docenti iscritti all'albo regionale
100	altro (specificare).

Per il trattamento della modalità *altro* si adotta la stessa tecnica prevista per *CodModalitaTemporali*

Relazione FontiFinanziarie

```
CREATE TABLE FontiFinanziarie(  
  IdCorso char (8) NOT NULL ,  
  CodOrigine smallint NOT NULL,  
  CodSpecifica smallint NOT NULL,  
  SpecificaDiAltro varchar (256) NULL,  
  CONSTRAINT PK_Fonti PRIMARY KEY CLUSTERED (  
    IdCorso,CodOrigine,CodSpecifica  
  )  
)
```

La proprietà *CodOrigine* assume i seguenti valori:

<i>CodOrigine</i>	
01	Fondi Comunitari
02	Fondi pubblici
03	Fondi privati
100	Altro

La proprietà *CodSpecifica* assume i seguenti valori:

<i>CodSpecifica (origine = Fondi Comunitari)</i>	
01	<i>FSE, obiettivi, assi, etc</i>
02	<i>Programmi e iniziative comunitarie</i>
03	<i>Fondi comunitari di altra natura</i>

<i>CodSpecifica Origine = Fondi pubblici</i>	
01	<i>Nazionali</i>
02	<i>Regionali</i>
03	<i>Provinciali</i>
04	<i>Comunali</i>
05	<i>Comunità Montana</i>

<i>CodSpecifica Origine = Fondi privati</i>	
01	<i>Associazioni imprenditoriali</i>
02	<i>Associazioni di categoria</i>
03	<i>Autofinanziamento</i>
100	<i>Altro</i>

Relazione CorsiMaster

```
CREATE TABLE CorsiMaster (  
  IdCorso char (8) NOT NULL ,  
  Titolo varchar (255) NOT NULL ,  
  Finalità text null,  
  OrganizzazioneInModuli char(1) NOT NULL,  
  PartecipazioneSingoliModuli char(1) NULL,  
  CodNaturaDelCorso smallint NULL,  
  CollaborazionePartnenariato text NULL,  
  MaxAmmessi smallint NULL,  
  QuotaIscrizione int NULL,  
  BorseDiStudio char(1) NULL,  
  ModalitaFruizioneBorsa text NULL,  
  Edizione smallint NULL,  
  CostoOraAllievo int NULL,  
  AltreCondizioniAccesso text NULL,  
  ModalitaAmmissione text NULL,  
  CodValutazioneCompetenze smallint NULL,  
  CodCertificazioneFinale smallint NULL,  
  CodModalitàDidatticaPrevalente smallint NULL,  
  CodModalitàTemporali smallint NULL,  
  CodDurata smallint NULL,  
  AnnualitàInCorso smallint NULL,  
  OreComplessive smallint NULL,  
  AltreInformazioni text NULL,  
  DataInserimento datetime NOT NULL,  
  DataUltimaModifica datetime NOT NULL,  
  UtentePrimoInserimento varchar (20) NOT NULL ,  
  UtenteProponente varchar (20) NOT NULL ,  
  UtenteApprovante varchar (20) NOT NULL ,  
  NoteAmministratore text NULL ,  
  CONSTRAINT PK_CorsiMaster PRIMARY KEY CLUSTERED (  
    IdCorso  
  )  
)
```

Nella relazione *CondizioniDiAccesso*, per questi corsi la proprietà *CodRequisito* assume i seguenti valori:

CodRequisito	
05	Diploma scuola secondaria
06	Diploma universitario
07	Laurea
100	Altro

CodCertificazione finale	
01	Frequenza
02	Qualifica
03	Specializzazione
04	Diploma
05	Nessuna
100	Altro

Relazione ModuliMaster

```
CREATE TABLE ModuliMaster (  
    IdCorso char (8) NOT NULL ,  
    NumProgressivo smallint NOT NULL,  
    Titolo varchar(255) NULL,  
    Ore smallint NULL,  
    Dal datemine null,  
    Al datetime null,  
    CONSTRAINT PK_ModuliMaster PRIMARY KEY CLUSTERED (  
        IdCorso , NumProgressivo  
    ),  
    CONSTRAINT FK_ModuliMaster FOREIGN KEY (  
        IdCorso  
    ) REFERENCES dbo.CorsiMaster (  
        IdCorso  
    )  
)
```

Relazione CorsiIFTS

```
CREATE TABLE CorsiIFTS (  
    IdCorso char (8) NOT NULL ,  
    Titolo varchar (255) NOT NULL ,  
    FiguraProfessionale text NULL,  
    SbocchiOccupazionali text NULL,  
    QuotaIscrizione int NULL,  
    MaxAmmessi smallint NULL,  
    CostoOraAllievo int NULL,  
    OreComplessive smallint NULL,  
    AgevolazioniFinanziarie text NULL,  
    DestinatariCondizioniAccesso text NULL,  
    ModalitàDiAmmissione text NULL,  
    CodValutazioneCreditiIniziale smallint NULL,  
    CodValutazioneCreditiFinale smallint NULL,  
    PianiIndividualizzabili char(1) NULL,  
    Attivato char(1) NOT NULL,  
    CertificazioneIntermedia varchar(255) NULL,  
    CertificazioneFinale varchar(255) NULL,  
    NumeroSemestri smallint NULL,  
    NumeroModuli smallint NULL,  
    CodModalitàTemporali smallint NULL,  
    AltreInformazioni text NULL,  
    DataInserimento datetime NOT NULL,  
    DataUltimaModifica datetime NOT NULL,  
    UtentePrimoInserimento varchar (20) NOT NULL ,  
    UtenteProponente varchar (20) NOT NULL ,  
    UtenteApprovante varchar (20) NOT NULL ,  
    NoteAmministratore text NULL ,  
    CONSTRAINT PK_CorsiIFTS PRIMARY KEY CLUSTERED (  
        IdCorso  
    )  
)
```

OreComplessive non sempre è ricavabile dalla somma delle ore per semestre, in quanto non sempre viene fornito tale dettaglio, per cui occorre fornirlo.

NumeroModuli e *NumeroSemestri* sono uguali al numero dei moduli/semestri presenti nelle relazione *SemestriIfTs* e *ModuliIFTs*, non possono essere campi calcolati, perché spesso viene solo fornito il numero di essi senza i dettagli.

CodValutazioneCreditiIniziale	
01	Iniziale, per l'accesso al corso

Questa codifica si poteva evitare in quanto descrive solo la presenza della valutazione, per cui è sufficiente dichiararlo char (1). Viene lasciato come codifica per uniformità di trattamento e per consentire ulteriori eventuali codifiche.

CodValutazioneCreditiFinale	
01	Finale con riconoscimento certificato
02	Finale senza riconoscimento certificato

La relazione non è normalizzata per semplificazione di implementazione e perché le codifiche di valutazione sembrano stabili.

Relazione ServiziSupporto

```
CREATE TABLE ServiziSupporto (
    IdCorso char (8) NOT NULL ,
    CodServiziDiSupporto smallint NULL,
    CONSTRAINT PK_ServiziSupportoIFTs PRIMARY KEY CLUSTERED (
        IdCorso , CodServiziDiSupporto
    )
)
```

Questa relazione contiene i servizi di supporto per IFTS.

CodServiziDiSupporto	
01	Informativo/orientativo
02	Psico-motivazionale
03	Professionale (tecnologico)

Relazione SemestriIFTs

```
CREATE TABLE SemestriIFTs (
    IdCorso char (8) NOT NULL ,
    Semestre smallint NOT NULL,
    Titolo varchar(255) NULL,
    Ore smallint NULL,
    CONSTRAINT PK_SemestriIFTs PRIMARY KEY CLUSTERED (
        IdCorso , Semestre
    ),
    CONSTRAINT FK_SemestriIFTs FOREIGN KEY (
        IdCorso
    ) REFERENCES dbo.CorsiIFTs (
        IdCorso
    )
)
```

Ore non sempre è ricavabile dalla somma delle ore dei moduli per semestre, in quanto non sempre viene fornito tale dettaglio, per cui occorre fornirlo

Relazione ModuliIFTS

```
CREATE TABLE ModuliIFTS (  
    IdCorso char (8) NOT NULL ,  
    Semestre smallint NOT NULL,  
    NumeroModulo smallint NOT NULL,  
    Titolo varchar(255) NULL,  
    Ore smallint NULL,  
    Dal datetime NULL,  
    Al datetime NULL,  
    CONSTRAINT PK_ModuliIFTS PRIMARY KEY CLUSTERED (  
        IdCorso , Semestre, NumeroModulo  
    ),  
    CONSTRAINT FK_ModuliIFTS FOREIGN KEY (  
        IdCorso , Semestre  
    ) REFERENCES dbo.SemestriIFTS (  
        IdCorso , Semestre  
    )  
)
```

Bibliografia

- N. Aloia, C. Concordia F. Furfari: Technologies for Public Acces Web Information Systems*, EuroWeb2001, Pisa 2001.
- N. Aloia, C. Concordia, F. Furfari, V. Miori: Caratteristiche, problematiche e tecnologia dei sistemi informativi basati sul Web*, Rivista di Informatica AICA, vol. XXXI n.2 settembre 2001 (pp. 95-108)
- N. Aloia, C. Concordia, F. Furfari, A Public Access Web Information System*, Proc. of 2001 International Conference on Intelligent Agents, Web Technologies and Internet Commerce, July 2001, Las Vegas
- N. Aloia, C. Concordia, V.Miori: Web Based Information Systems*, Proc. of 16th IFIP World Computer Congress 2000, Information Technology for Business Management, August 2000, Beijing, (pp. 581-588)
- N. Aloia, C. Concordia, F. Furfari, V.Miori: Considerazioni per la realizzazione di sistemi informativi basati su Web accessibili ai disabili*, Proc. of 6th Convegno Nazionale Informatica, Didattica e Disabilità – Andria (BA), novembre 1999.
- N. Aloia, C. Concordia, V.Miori: Web Architectures for Database Access*, Proc. of WebNet '99 - Honolulu, Hawaii, ottobre 1999, (pp. 1473-1475)
- N. Aloia, C. Concordia, V.Miori: INFEA: un sistema informativo per il Ministero dell'Ambiente*, Proc. of AICA '99 (pp. 447-452), Abano Terme, settembre 1999
- N. Aloia, M. Matera, F. Paternò: Presentations for databases in multimedia environments, Multimedia Systems* – ACM/Springer ed., Multimedia Systems Journal vol. 6(6): 408-420 (1998)
- S. Tardelli, L. Versienti Architettura software, funzionalità e meccanismi del Sistema INFEA*, Rapporto CNUCE-B4-2002-004
- S. Tardelli, L. Versienti Il sistema informativo per il Servizio Sviluppo Sostenibile del Ministero dell'ambiente: Manuale di utilizzo dell'interfaccia per utenti esperti*, Rapporto CNUCE-B4-2002-003

Indice

Introduzione.....	1
1. Progettazione concettuale della base di dati	2
Classe Ente.....	7
Classe Locazione.....	12
Classe Offerta.....	12
Classe Offerta EA.....	13
Classe Richiesta Finanziamenti.....	18
Classe Materiali.....	20
Classe OffertaFormativa (Corsi).....	21
2. Progettazione logica e fisica della base di dati	36
2.1 Trasformazione dello schema concettuale.....	37
2.1.1 Trasformazione della classe Ente.....	37
Relazione Enti.....	38
2.1.2 Trasformazione delle associazioni che coinvolgono la classe Ente.....	42
Relazione OffertaEnte.....	42
Relazione FunzioniEA.....	47
Relazione Servizi.....	48
Relazione DocumentiIstitutivi.....	49
2.1.3 Trasformazione della classe Offerta.....	49
Relazione Offerta.....	51
Relazione Iniziativa.....	51
Relazione Visita/Soggiorno.....	52
Relazione Esperienza.....	53
Relazione PeriodoDiSvolgimento.....	54
Relazione Progetti.....	54
Relazione Manifestazione.....	55
Relazione EventiSEA.....	56
Relazione ContestoAmbientale.....	57
Relazione FinalitaEducativa.....	57
Relazione DestinatariEA.....	58
Relazione Argomenti.....	58
2.1.4 Trasformazione della classe Materiali.....	59
Relazione Materiali.....	59
Relazione Autori.....	60
Relazione MaterialiEnti.....	61
Relazione OffertaMateriali.....	61
2.1.5 Trasformazione della classe Locazione.....	61
Relazione Locazione.....	62
Relazione ContestoTerritoriale.....	63
Relazione Regioni.....	63
Relazione Provincie.....	63
2.1.6 Trasformazione della classe Agenda.....	64
Relazione Agenda.....	64
Relazione Segnalazioni.....	65
2.2. Trasformazione della classe Utenti e gruppi.....	65
2.3 Trasformazione della classe Offerta Formativa.....	69
Relazione AreeTematiche.....	70
2.3.1 Trasformazione della classe Corso Universitario.....	71
Relazione CorsiUniversitari.....	71
2.3.2 Trasformazione della classe Corso Istruzione.....	72
Relazione CorsiIstruzioneSecondaria.....	73
Relazione CorsiADistanza.....	74
Relazione ModuliFAD.....	74
Relazione MaterialiDidatticiFAD.....	74
Relazione DotazioniFAD.....	75
Relazione DidatticaFAD.....	75
Relazione InterazioneCorsiADistanza.....	76

2.3.3 Trasformazione della classe Corso Formazione/IFTS/Master	77
Relazione CorsiDiFormazione.....	78
Relazione SedeDiSvolgimento	79
Relazione CondizioniDiAccesso	80
Relazione DestinatariFormazione.....	80
Relazione SpecificaDidattica.....	81
Relazione Docenza	82
Relazione FontiFinanziarie.....	83
Relazione CorsiMaster	84
Relazione ModuliMaster	85
Relazione CorsiIFTS	85
Relazione ServiziSupporto	86
Relazione SemestriIFTS	86
Relazione ModuliIFTS	87
Bibliografia.....	88
Indice	89