

INGEGNERIA E SCIENZE INFORMATICHE

UNIBO, Cesena (FC)

Museums Manager

Gestionale per l'amministrazione interna di musei e famiglie di musei

Esame di Basi di Dati 2020-21

Candidati:

Farneti Andrea (mat. 0000832033)

Pantieri Andrea (mat. 0000825053)

Sommario

1.

ANALISI DEI REQUISITI

Intervista al committente (manager di diversi musei sparsi per l'Italia):

“I nostri musei necessitano di un sistema gestionale per gli amministratori che permetta l’inserimento e la gestione di ognuno di essi, e l’archivio di ogni informazione ritenuta utile ed importante.

Le funzionalità principali che riteniamo essenziali sono la gestione di musei e dipendenti, l’elenco di ogni oggetto esposto all’interno del museo e relative informazioni, ed il calcolo di saldo e flusso per ogni museo.

È necessario che il sistema permetta anche la creazione e modifica di biglietti d’ingresso per un museo.”

Individuiamo le parole chiave da questa prima intervista:

*“I nostri musei necessitano di un sistema **gestionale** per gli amministratori che permetta **l’inserimento** e la gestione di ognuno di essi, e **l’archivio** di **ogni informazione** ritenuta utile ed importante.*

*Le funzionalità principali che riteniamo essenziali sono la **gestione di musei e dipendenti**, l’elenco di **ogni oggetto esposto** all’interno del museo **e relative informazioni** ed il **calcolo di saldo e flusso** per ogni museo.*

*È necessario che il sistema permetta anche la **creazione e modifica** di biglietti d’ingresso per ogni museo.”*

La prima intervista è stata troppo vaga e ci ha lasciato con diverse domande:

- Quali sono i criteri secondo i quali un’informazione è ritenuta importante?
- Cosa si intende per gestione di musei e dipendenti nello specifico?
- Di quali informazioni dobbiamo prendere nota per ogni oggetto esposto?
- Ogni quanto è necessario calcolare saldi e flussi dei musei?
- In che livello di dettaglio bisogna scendere per musei, ed oggetti?
- Come funzionano i biglietti nello specifico?

Abbiamo richiesto una seconda intervista al committente per delucidazioni:

“Il servizio di gestione di ogni museo e dipendente deve permettere l’aggiunta, la modifica e l’eliminazione di ognuno di essi.

Informazioni ritenute importanti per noi sono le varie statistiche prodotte da ogni museo, da archiviare sia mensilmente che annualmente. Per ogni oggetto esposto all’interno di un museo è tassativamente necessario conoscerne provenienza, periodo storico ed autore/i. Un oggetto può essere composto da più parti, per cui potrebbe essere interessante disporre di tali informazioni anche per le singole parti di cui si compone un oggetto. Per quanto riguarda i musei, essi potrebbero contenere delle sezioni e, a loro volta, delle sottosezioni. I dipendenti, invece, devono essere registrati in base al loro ruolo ed in base all’orario di arrivo ed uscita dal lavoro. Ogni biglietto dovrà avere una data di validità e un proprio prezzo, oltre che una tipologia (sconto studente, bambini...). Inoltre, voglio che i clienti possano vedere le date in cui un museo è chiuso o aperto specialmente, in modo che possano comprare i biglietti di conseguenza. Il numero di biglietti totale potrebbe anche variare se è un giorno di apertura speciale. Un museo può offrire ai visitatori diversa scelta riguardo ai biglietti d’entrata.”

Individuiamo ora nuovamente le parole chiave:

“Il servizio di gestione di ogni museo e dipendente deve permettere l’aggiunta, la modifica e l’eliminazione di ognuno di essi.

Informazioni ritenute importanti per noi sono le varie statistiche prodotte da ogni museo, da archiviare sia mensilmente che annualmente. Per ogni oggetto esposto all’interno di un museo è tassativamente necessario conoscerne provenienza, periodo storico ed autore/i. Un oggetto può essere composto da più parti, per cui potrebbe essere interessante disporre di tali informazioni anche per le singole parti di cui si compone un oggetto. Per quanto riguarda i musei, essi potrebbero contenere delle sezioni e, a loro volta, delle sottosezioni. I dipendenti, invece, devono essere registrati in base al loro ruolo ed in base all’orario di arrivo ed uscita dal lavoro. Ogni biglietto dovrà avere una data di validità e un proprio prezzo, oltre che una tipologia (sconto studente, bambini...). Inoltre, voglio che i clienti possano vedere le date in cui un museo è chiuso o aperto specialmente, in modo che possano comprare i biglietti di conseguenza. Il numero di biglietti totale

potrebbe anche variare se è un giorno di apertura speciale. Un museo può offrire ai visitatori diversa scelta riguardo ai biglietti d'entrata.”

[Si ipotizza sia avvenuta anche una terza ed ultima intervista chiarificativa col committente nella quale egli ha specificato nel dettaglio di quali caratteristiche è composta ogni “entità” definita nella seconda intervista. Ad esempio: “per ogni museo bisognerà inserire gli orari di apertura/chiusura, il numero di biglietti massimo...”)]

Una volta individuati ed estratti i concetti principali dall’intervista col committente stiliamone un documento dettagliato riassuntivo che includa tutti i requisiti di cui sarà composto l’applicativo e che dovranno necessariamente essere implementati (*specifica dei requisiti*):

L’applicativo da implementare è un gestionale per musei rivolto unicamente al personale di amministrazione, in altre parole ai dirigenti/capi dei musei stessi. Esso deve permettere la realizzazione delle seguenti funzionalità (per le quali è necessario un database):

- Creazione, modifica ed eliminazione di un museo (con relative informazioni di base quali nome, luogo, orario di apertura/chiusura e numero di biglietti massimo)
- Creazione di nuove famiglie di musei (gruppi di musei correlati tra loro per i quali si vogliono ottenere statistiche di gruppo; Es. Isola dei musei di Berlino)
- Creazione, modifica ed eliminazione di specifiche sezioni (ed eventuali sottosezioni) di cui è composto un museo. Esempio: la macro-sezione di Storia di un museo potrebbe essere composta dalle sottosezioni di storia romana, storia etrusca ecc...
- Aggiunta, modifica ed eliminazione di contenuti all’interno di un museo (e degli eventuali sottocontenuti di cui è composto), posizionati all’interno di una sezione del museo. Ogni contenuto disporrà di diverse informazioni quali nome, descrizione, data di ritrovamento e di arrivo al museo, provenienza, creatore e periodo storico. Esempio: il modellino di una bomba atomica contenuto nella sezione di Scienza di un museo è formato a sua volta dalla scocca, il detonatore, l’esplosivo... Anche per ognuna di queste sottoparti si vogliono avere delle informazioni.
- Deve essere possibile cercare i contenuti di un museo per provenienza, tipo, creatore e periodo storico.

- Aggiunta, modifica ed eliminazione di vari tipi di contenuto. Esempio: la sezione di Storia di un museo potrebbe contenere, fra i vari oggetti, una serie di affreschi. Può essere comodo quindi ricercare i contenuti di un museo per tipo (nell'esempio, tipo = affresco).
- Aggiunta, modifica ed eliminazione degli autori che hanno realizzato determinati contenuti (nome, cognome, anno di nascita, descrizione).
- Aggiunta, modifica ed eliminazione del personale che lavora all'interno di un museo (bisogna conoscerne nome, cognome, numero di cellulare/indirizzo mail e lo stipendio orario)
- Funzionalità di registro presenze che permetta di registrare la data di entrata/uscita dei dipendenti dal lavoro.
- Funzionalità di registro manutenzioni che permetta di tenere conto di tutte le manutenzioni che sono state fatte per un determinato museo (la data della manutenzione, ciò che è stato fatto ed il costo)
- Creazione, modifica ed eliminazione di biglietti per l'accesso ad un museo. Ogni biglietto avrà una data di validità, un prezzo e potrà essere di un determinato tipo (esempio: ingresso studente, ingresso bambino...)
- Aggiunta ed eliminazione dei giorni di chiusura.
- Aggiunta, modifica ed eliminazione dei giorni di apertura speciali ed eventuali orari speciali di apertura/chiusura.
- Calcolo di tutte le statistiche mensili ed annuali di ogni museo e di ogni famiglia di musei (spese totali, fatturato, numero di biglietti venduti, numero di manutenzioni svolte, numero di nuovi contenuti aggiunti e numero di giorni di chiusura totali).

Ciò di cui non deve occuparsi questo gestionale riguarda:

- Come il personale compila il registro presenze (si ipotizza che esista un altro applicativo per dipendenti che utilizza la stessa sezione di database e tramite il quale essi compiono giornalmente check-in e check-out)
- Come i clienti acquistano i biglietti (si ipotizza che vengano stampati e comprati direttamente dentro il museo, oltretutto esista un applicativo/sito web che interagisca con la stessa sezione di database e

permetta di comprare/prenotare biglietti online registrandone le vendite).

1.

PROGETTAZIONE CONCETTUALE

A seguito della fase di analisi dei requisiti vengono progettati e proposti i seguenti schemi scheletro per definire la struttura dell'applicativo:

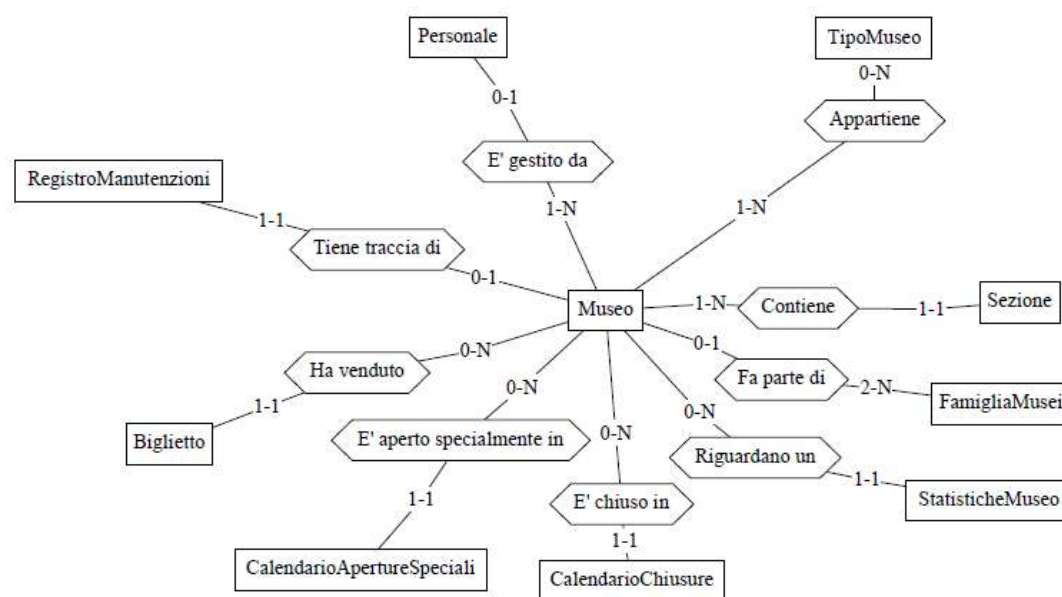
Museo:

L'entità “**Museo**” rappresenta il fulcro dell'intero schema E/R ed è l'entità principale sulla quale si basa il progetto. Ogni entità associata ad essa rappresenta, nella pratica, una funzionalità che il museo svolge e che l'applicativo deve permettere di realizzare.

È possibile suddividere queste funzionalità in due macro-argomenti:

- la gestione del personale, dei clienti e delle statistiche
- la gestione dei contenuti interni al museo

In seguito sarà necessario convertire la relazione “*Appartiene*” associata a “**Museo**” e “**TipoMuseo**” in un'entità, in quanto un museo può essere di più tipi e più tipi di museo potrebbero ripetersi fra più musei (di conseguenza per individuare univocamente un museo ed il suo tipo (senza duplicare i record all'interno di TipoMuseo che creerebbe ridondanza) è necessario l'utilizzo di una terza entità).



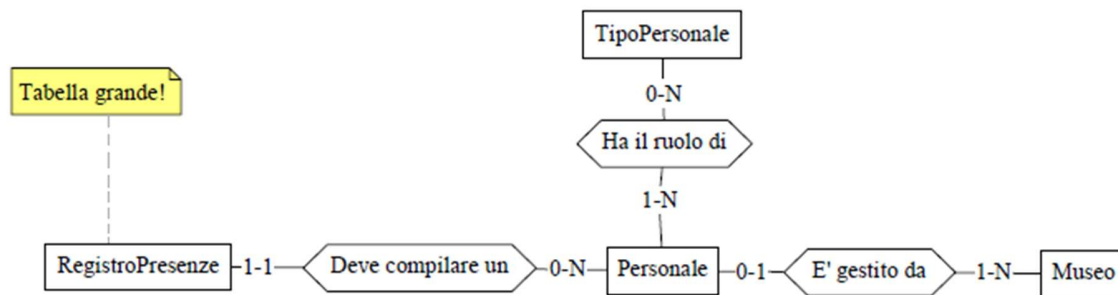
Personale:

Dato che l'applicativo che ci è stato commissionato non necessita di definire entità specifiche di personale con caratteristiche e ruoli univoci (ad esempio manager, dipendente, guida...) non è stato necessario realizzare una generalizzazione gerarchica con delle entità specificative, ma è stato sufficiente definire l'entità **"TipoPersonale"**, associata a **"Personale"** dalla relazione **"Ha ruolo di"**.

Adottando questa soluzione però, si nota che sarà necessario trasformare quest'ultima relazione in un'entità, in quanto più ruoli di lavoro potrebbero ripetersi fra più persone e una persona potrebbe avere più ruoli.

Chiaramente il personale varierà a seconda del museo in cui lavora e ognuno di essi dovrà effettuare giornalmente l'operazione di check-in e check-out a lavoro, che verranno salvate nell'entità **"RegistroPresenze"**.

Dato che l'applicativo da realizzare è rivolto solamente agli amministratori del museo e non ai lavoratori, **"RegistroPresenze"** è utilizzato all'interno dell'applicativo solo in lettura. Esso non si occupa di come la futura tabella verrà compilata (si ipotizza esista un sistema o un applicativo esterno dedicato al personale tramite il quale vengono registrate le presenze sul luogo di lavoro e salvate come record in **"RegistroPresenze"**).



Si noti che "RegistroPresenze" risulterà essere una tabella di grandi dimensioni, in quanto dovrà ricevere giornalmente nuovi record per ogni personale presente a lavoro in ogni museo; sarà quindi necessario provvedere nella fase di progettazione logica ad ottimizzarne la sua gestione tramite diverse tecniche, a seconda delle operazioni che dovrà svolgere (select, insert into..).

//TABELLA DA INDICIZZARE PER LE OPERAZIONI DI RICERCA ED ELIMINARE I RECORD ANNUALMENTE (E SALVARLI IN UN ARCHIVIO)

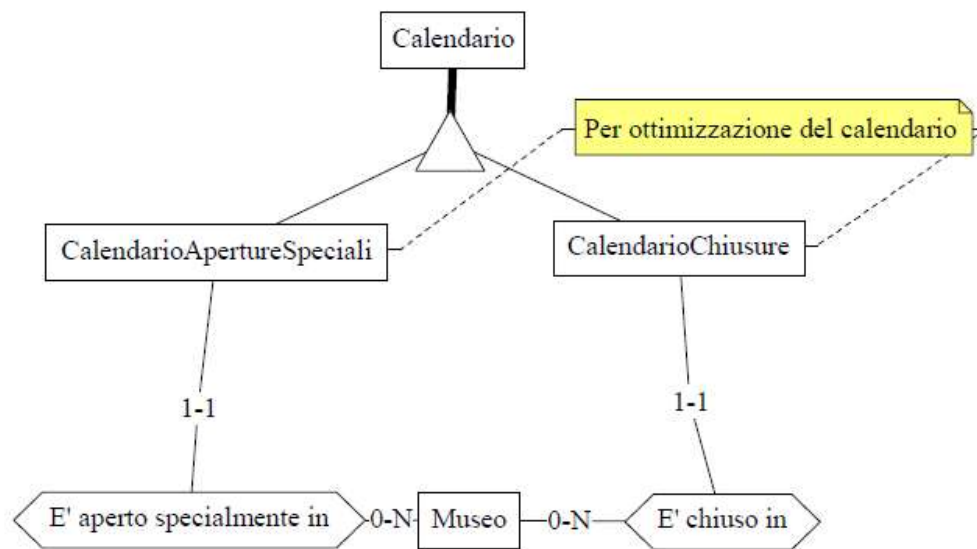
Calendario:

Un'altra funzionalità richiesta per questo applicativo è la gestione delle giornate di apertura/chiusura e dei relativi orari di ogni museo. Questa funzionalità risulta essere inoltre necessaria per permettere ai clienti (tramite un applicativo esterno) di visionare le date di apertura di un museo, permettendo quindi l'acquisto di biglietti. Per realizzare tali operazioni sarà quindi necessaria la creazione di un sistema di calendario (variabile, ovviamente, per ogni museo).

Per stabilire sia i giorni di apertura che quelli di chiusura abbiamo quindi creato ed organizzato una gerarchia per "Calendario" in modo tale da ottimizzarlo, rimuovendo la necessità di salvare ed aggiornare periodicamente lo stato di tutti i giorni annuali (questa affermazione è valida finché ogni museo mantiene un numero maggiore di giorni di apertura rispetto a quelli di chiusura).

Questa gerarchia sarà composta da due figli:

- **CalendarioApertureSpeciali:** identifica i giorni di apertura speciale, come i giorni festivi oppure giornate con orari speciali
- **CalendarioChiusure:** indica tutti i giorni di chiusura.



Biglietto:

“**Biglietto**” non è altro che un’entità che ha l’obiettivo di archiviare gli acquisti effettuati dai clienti a fini statistici. Dato che l’applicativo che ci è stato commissionato è rivolto solamente al personale di amministrazione, esso non dovrà occuparsi di fornire un’interfaccia ai clienti per l’acquisto di biglietti, ma il database deve comunque essere realizzato in modo tale da permettere anche tale funzione (perché si presuppone che esista un applicativo esterno che si conetterà e utilizzerà la stessa porzione di database, salvando gli acquisti in “Biglietto”).

Il nostro applicativo utilizzerà quindi “Biglietto” in sola modalità “lettura”.

Specifichiamo ora tre limitazioni che abbiamo deciso di imporre a carattere generale ai biglietti e al calendario per semplificare il dominio del problema:

- Un biglietto è valido per una sola persona.
- Un biglietto è valido per una sola giornata.
- Il calendario non prevede come impostare un periodo di chiusura per un museo (è solo possibile impostare ogni giorno di chiusura singolarmente)

Abbiamo deciso di imporre tali limitazioni onde evitare in “Calendario” anche la definizione di un sistema di “range temporale” per gestire solamente questi ultimi

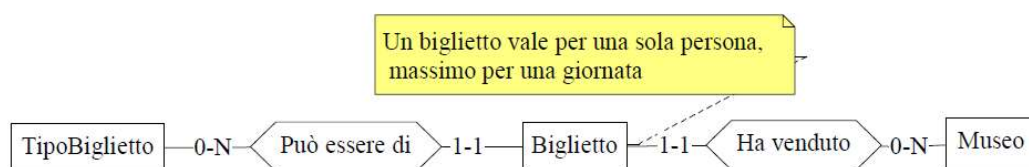
due casi specifici (ovvero biglietti validi per più giornate ed il poter impostare periodi di chiusura): il tutto sarebbe risultato ridondante per la data ed eccessivo come carico di lavoro.

È stato necessario anche modellare come gestire più tipologie di biglietti; Uno studente, ad esempio, potrebbe avere diritto ad uno sconto sul prezzo del biglietto rispetto ad un adulto.

Di seguito ne verranno indicate solo alcune delle possibili tipologie esistenti:

- Studente (prezzo 70%);
- Biglietto normale (over 25, under 65);
- Under 25, over 15 (prezzo 75%);
- Under 15 (prezzo 50%);
- Over 65 (prezzo 50%).
- ...

Abbiamo deciso quindi di non utilizzare una gerarchia per i tipi di biglietti affinché non ne sia limitato il numero massimo di varietà.

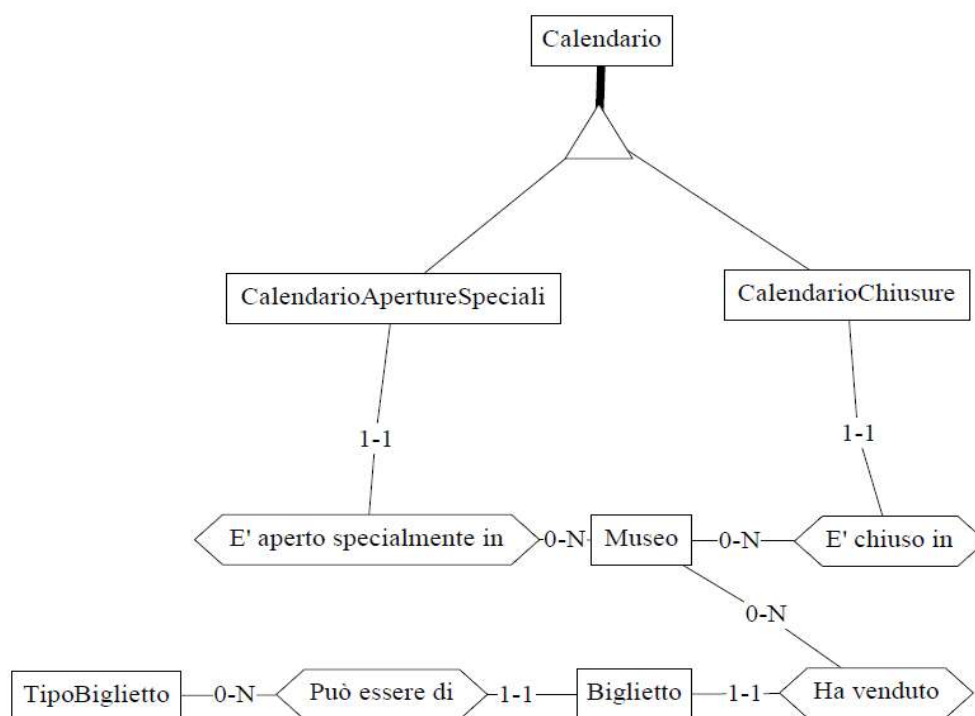


Relazione fra Calendario e Biglietto:

Il seguente schema scheletro mostra l'interazione che esiste fra la funzionalità di calendario e la funzionalità di vendita biglietti. È già stato spiegato come sia l'acquisto di biglietti che la popolazione della (futura) relativa tabella "**Biglietto**" non vengano gestiti all'interno di questo applicativo ma esternamente, tuttavia, il database è stato progettato in modo tale da permettere il corretto funzionamento anche di tali funzionalità (supponendo quindi che il servizio esterno si colleghi a questo database); proprio per questo è importante la relazione fra "**Biglietto**" e "**Calendario**": tramite quest'ultimo è possibile risalire alle giornate di apertura di

un museo, mentre tramite “**TipoBiglietto**” è possibile selezionare il biglietto desiderato. Con queste informazioni è poi possibile per il cliente procedere con l’acquisto, che verrà infine salvato in “**Biglietto**” con data di validità selezionata (ricordando che un biglietto per un museo ha validità di un solo giorno).

Inoltre, tramite questo tipo di relazione è possibile specificare anche il numero di biglietti massimi giornalieri per ogni circostanza, ovvero a seconda che sia per un museo un giorno di apertura normale che un giorno di apertura speciale.



Statistiche:

L’elaborazione delle statistiche rappresenta forse la parte più significativa ed importante che deve svolgere l’applicativo: esse rappresentano l’output del programma e permettono di ottenere dati di importanza economica riguardo a tutte le informazioni che gli amministratori inseriscono in input.

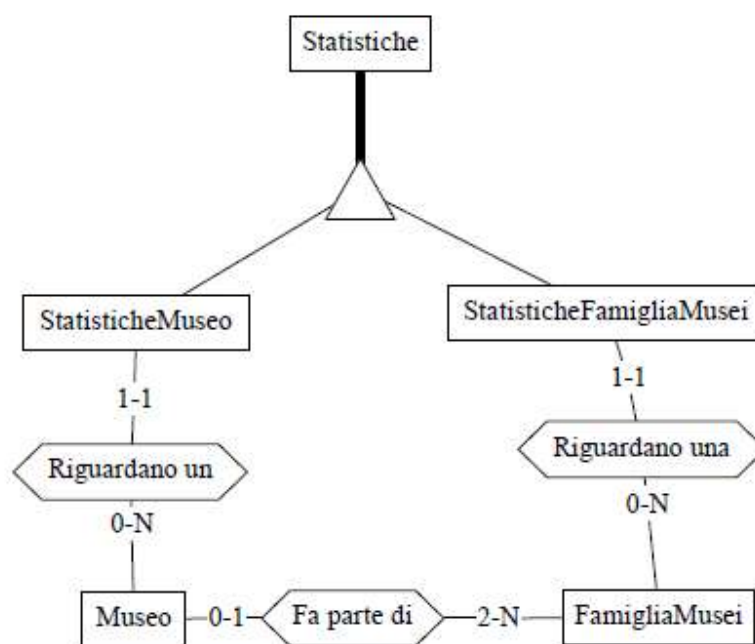
L'applicativo deve permettere di fornire dati mensili e annuali periodici per ogni museo, oltre che per ogni famiglia di musei.

Le statistiche fra un museo e una famiglia di musei sono le stesse, la differenza sta solamente nel fatto che ogni singola statistica di una famiglia di musei è ottenuta dalla somma totale della stessa di ogni singolo museo, per ogni museo che compone la famiglia.

Ad esempio: la statistica “fatturato” della famiglia di musei “A” (composta da 3 musei) sarà data dalla somma della statistica “fatturato” del museo 1, del museo 2 e del museo 3.

Deve però anche essere possibile visionare le statistiche di ogni singolo museo che compone la famiglia singolarmente.

Per ottenere ciò abbiamo realizzato il seguente schema scheletro:

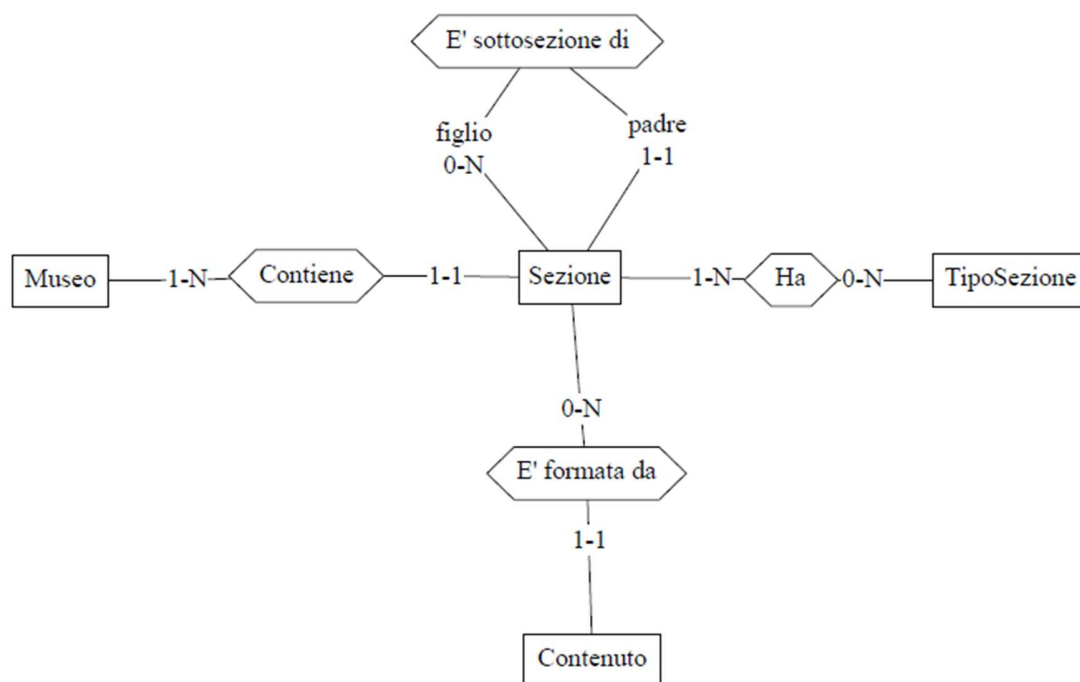


Sezione:

Ciascun museo può essere diviso in varie sezioni (“**Sezione**”), da quella principale ad altre di diverso tipo (storia, cultura, arte...). L’elaborazione di queste è stata gestita nella maniera più generale possibile in modo da includere diverse possibilità e combinazioni; Ciascuna sezione del museo potrà contenere, infatti, diverse sottosezioni (“**Sottosezione**”) (gestito tramite l’associazione “*E’ sottosezione di*”) e così via continuando in modo ricorsivo. In aggiunta, ognuna di

queste apparterranno a più tipi di sezioni (“**TipoSezione**”), ad esempio “Sezione di storia e arte” sarà sia di tipo storia che di tipo arte, mentre le due singole sottosezioni di cui è formata saranno relativamente di tipo storia, e di tipo arte.

L’applicativo deve infine permettere di catalogare all’interno di ogni “Sezione” i contenuti (“**Contenuto**”) sparsi all’interno di tutte le aree del museo.



Contenuto:

Ultimo aspetto da trattare nello specifico è quello del contenuto.

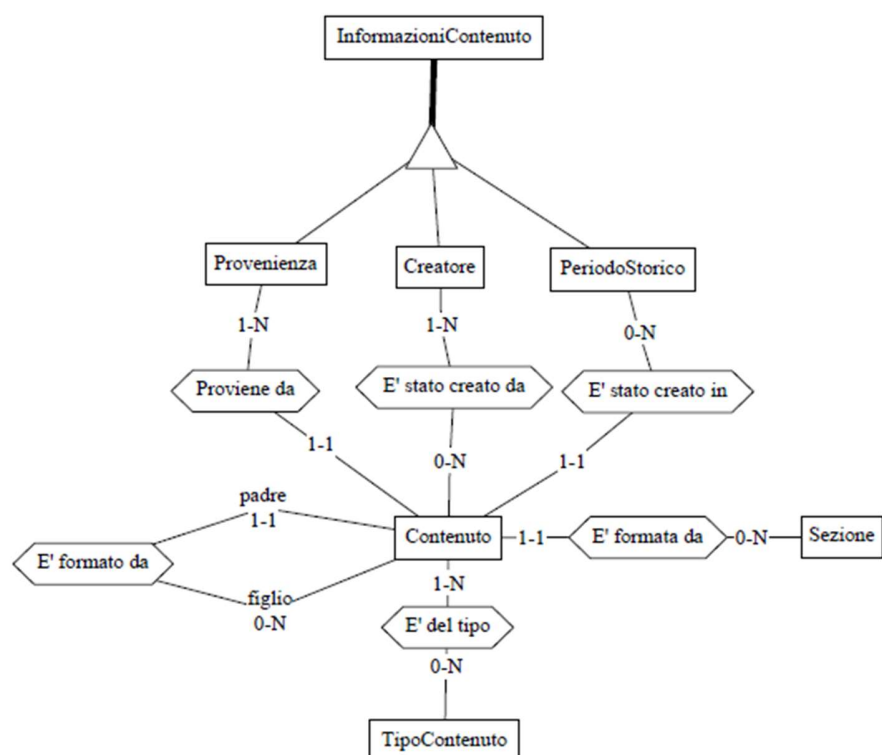
Come richiesto dal committente, un contenuto può essere formato da più parti e, a loro volta, ognuna di esse può essere formata da altre e così via: per l’implementazione di tale funzionalità è stata sufficiente una relazione unaria a “**Contenuto**” con cardinalità 1-1 per il padre e 0-N per il figlio: in questo modo, se uno più figli esistono (0-N, ovvero uno o più sottocontenuti) essi avranno necessariamente uno e uno solo padre (1-1, ovvero il contenuto del livello immediatamente sovrastante).

Inoltre, come nel caso di “**Sezione**” e “**TipoSezione**”, per le stesse ragioni e per il fatto che all’interno dell’applicativo dovrà essere possibile eseguire ricerche per tipo, ogni “Contenuto” avrà un “**TipoContenuto**”.

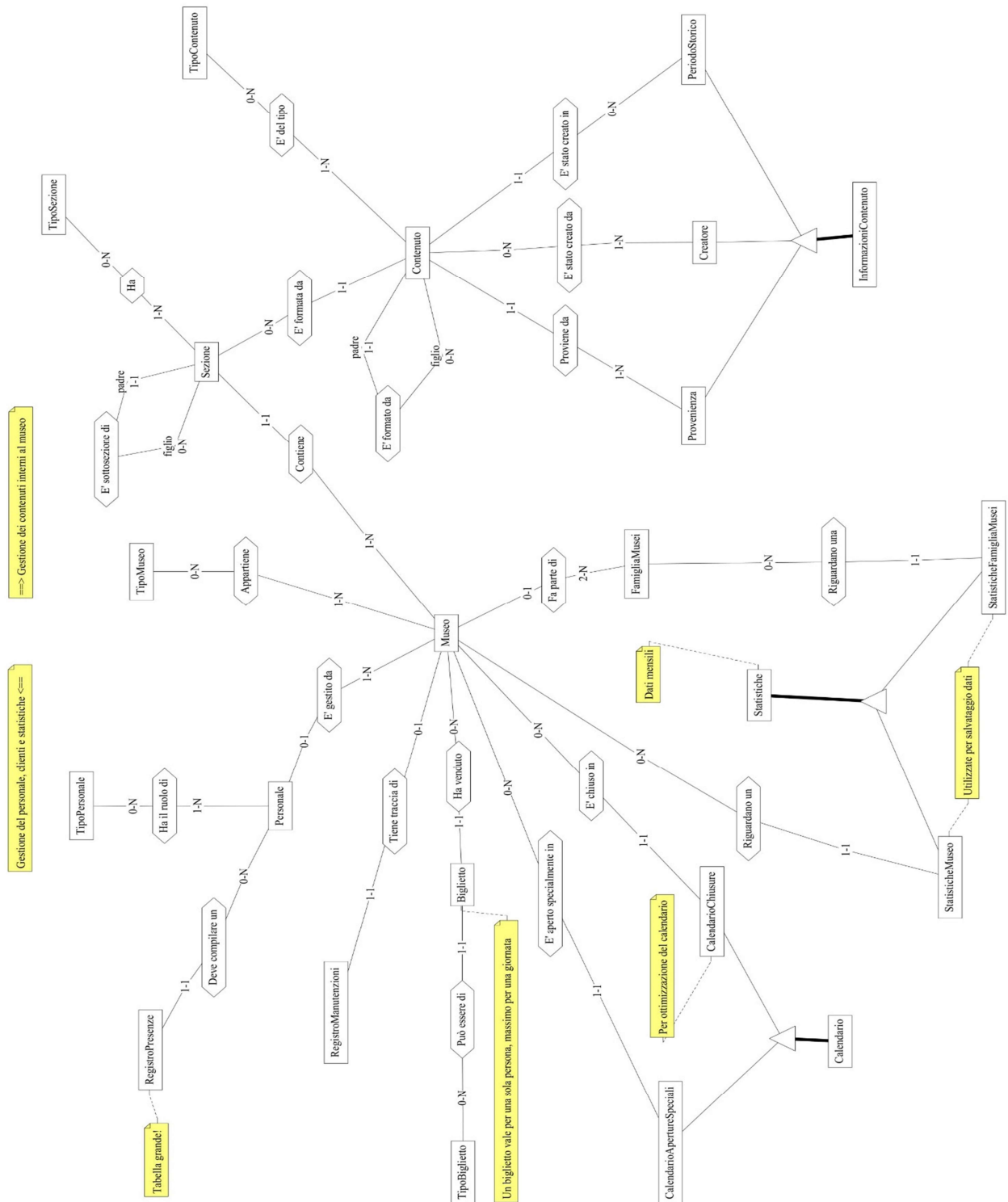
Per finire, oltre agli attributi di base che un contenuto può avere, il committente ha richiesto 3 specifiche informazioni complesse che ogni contenuto deve possedere e tramite cui si deve poter eseguire ricerche di contenuti all’interno del programma:

- “**Provenienza**”: definisce il luogo di origine del contenuto.
- “**Creatore**”: indica chi ha creato il contenuto; Nota che un contenuto potrebbe avere più autori (0-N, 0 se non se ne conosce l’autore).
- “**PeriodoStorico**”: rappresenta il periodo storico del contenuto; Esempio: Medioevo, 500 d.C. - 1000 d.C.

Per la realizzazione di questa specifica abbiamo utilizzato una gerarchia per ottenere un livello di astrazione che ottimizzasse le informazioni evitando ridondanze, dato che avevamo bisogno di tre specifiche entità dedicate le quali hanno anche diverse caratteristiche in comune, e non di un particolare meccanismo di estendibilità per aggiungere nuove informazioni.

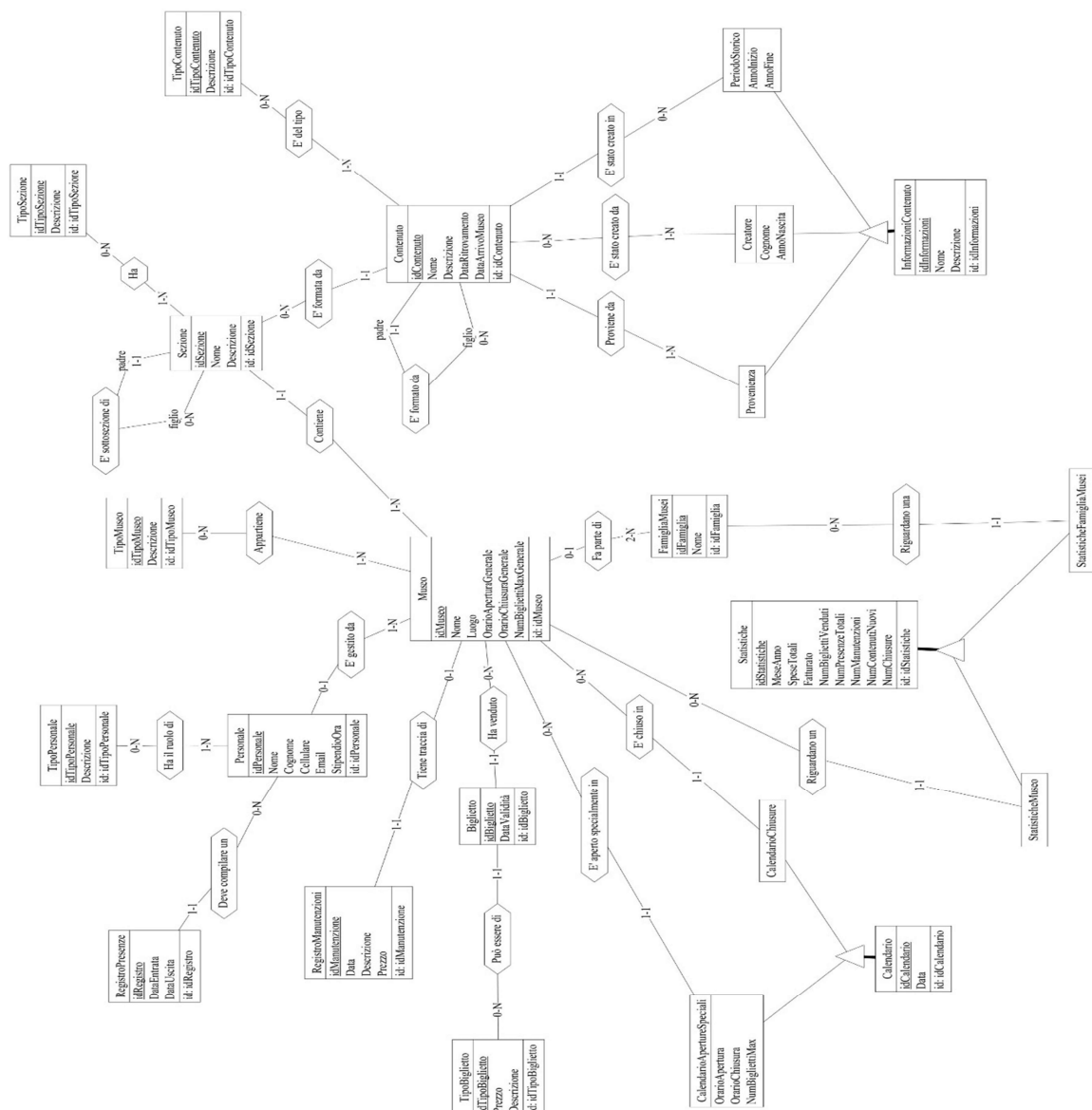


Schema scheletro completo (generico):



Schema scheletro completo (finale):

Dopo aver analizzato il dominio del problema ed aver definito la struttura di specifiche soluzioni (spiegate precedentemente in dettaglio) per ogni funzionalità definita nella specifica dei requisiti, prendendo anche già nota di possibili problemi di ottimizzazione che sono sorti legati al futuro (tramite le note in giallo, dei quali ce ne occuperemo nel loro contesto ovvero nella progettazione logica), abbiamo concluso la progettazione concettuale con il seguente schema E/R, aggiungendo ogni attributo necessario al corretto funzionamento di base dell'applicativo:



2.

PROGETTAZIONE LOGICA