Progetto di Basi di Dati 2

Centro commerciale

Francesco Pontillo

Matricola 600119

Sommario

1 Requisiti 1

1.1 Raccolta dei requisiti 1

1.2 Analisi dei requisiti 1

1.2.1 Livello di astrazione 1

1.2.2 Standardizzazione e linearizzazione 2

1.2.3 Omonimie e sinonimie 2

1.2.4 Requisiti trasformati 2

1.2.5 Raggruppamento di frasi per concetti 3

1.2.6 Glossario dei termini 4

2 Progettazione concettuale 5

2.1 Schema concettuale 5

2.1.1 Schema scheletro 5

2.1.2 Esplosione di entità 5

2.1.3 Schema concettuale finale 6

2.2 Dizionario dei dati 6

2.2.1 Entità 6

2.2.2 Relazioni 7

2.3 Regole aziendali 7

2.3.1 Regole di vincolo 7

2.3.2 Regole di derivazione 7

3 Progettazione logica 8

3.1 Operazioni 8

3.2 Ristrutturazione dello schema E-R 8

3.2.1 Tavola dei Volumi 8

3.2.2 Tavola delle Operazioni 9

3.2.3 Tavole degli Accessi 9

3.2.4 Analisi delle ridondanze 10

3.2.5 Eliminazione delle gerarchie 11

3.2.6 Partizionamento ed accorpamento di concetti 12

3.2.7 Scelta degli identificatori principali 13

3.3 Schema ER ristrutturato 13

3.4 Modello logico 14

4 Progettazione fisica 15

4.1 Tipi e tabelle 15

4.2 Indici 15

4.2.1 Indice b-tree su attivita.nome 15

4.2.2 Indice b-tree su dipendente.cognome e dipendente.nome 15

4.2.3 Indice hash su struttura.codice 15

4.2.4 Indice b-tree su tipo\_attivita.descrizione 15

4.2.5 Indice b-tree su tipo\_struttura.descrizione 15

4.2.6 Indice hash su user.username 15

4.2.7 Indice hash su user\_session.authcode 15

4.3 Trigger 15

4.3.1 Gestione della ridondanza sul numero dei dipendenti 15

4.4 Funzioni 15

4.4.1 Cerca dipendente 15

4.4.2 Esistenza di sessione 16

5 Servizi Web RESTful 17

5.1 Autenticazione e sicurezza 17

5.2 Cross Origin Resource Sharing 17

5.3 Risorse esposte 17

6 Web Application 18

6.1 Componenti MVC 18

6.1.1 Router 18

6.1.2 Controller 18

6.1.3 View 18

6.2 Comunicazione con il servizio RESTful 19

7 Progettazione del Data Warehouse 20

7.1 Progettazione concettuale 20

7.2 Progettazione logica 20

7.3 Implementazione 20

# Requisiti

## Raccolta dei requisiti

Si desidera creare un sistema informatico per un centro commerciale.

|  |  |
| --- | --- |
| Centro commerciale | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Il centro commerciale è costituito da un insieme di edifici nei quali si concentrano numerose attività commerciali, come quelle per la grande distribuzione organizzata, negozi specializzati, cinema e attività di ristorazione. L’accesso agli edifici che ospitano le diverse imprese commerciali è reso possibile da piccole vie e piazze, spesso al coperto (gallerie) aperte solo al traffico pedonale.  Tutte le strutture di cui è composto il centro (gli edifici, le piazze, le gallerie...) sono identificate univocamente da una stringa alfanumerica, composta da un carattere identificativo della tipologia di struttura (E per edificio, P per piazza, G per galleria ecc.) e da un numero. Le varie attività commerciali sono identificate univocamente dal codice della struttura nella quale sono collocate e dal proprio codice numerico.  Per ogni attività commerciale si deve, quindi, memorizzare il nome della struttura nel quale è collocata ed il piano (nel caso degli edifici). I negozi, inoltre, sono organizzati per categorie. Alcuni esempi di tipologie di negozi sono l’abbigliamento, l’arredo e l’informatica. Altre informazioni di interesse per le attività commerciali sono il nome (o ragione sociale), numero di dipendenti, la partita iva e il nome del proprietario, se l’attività non è in franchising o, in caso contrario, il nome del responsabile.  Anche le attività di ristorazione sono suddivise per tipologia, come i fast-food, i ristoranti tipici di una particolare nazione del mondo o area geografica (italiana o estera).  Il database, infine, tiene memoria di tutti i dipendenti del centro. Per ogni dipendente, si vuole conoscere, insieme ai dati anagrafici, anche la data di assunzione (ed eventualmente quella di licenziamento) e l’attività commerciale presso la quale solo stati assunti e il periodo di lavoro. |

## Analisi dei requisiti

Si analizzano i requisiti, chiarendo concetti ambigui, ponendo un adeguato livello di astrazione, standardizzando e linearizzando le frasi più complesse. Verranno risolte le ambiguità introdotte da eventuali omonimie e sinonimie, presentando i concetti in maniera strutturata.

Infine verrà introdotto un glossario finale dei termini.

Nell’analisi dei requisiti si utilizzeranno i numeri di riga come riportati al paragrafo 1.1.

### Livello di astrazione

Ai righi 3 e 17 per “attività di ristorazione” si fa riferimento a ristoranti.

Al rigo 4, per “imprese commerciali” si fa riferimento ad attività commerciali.

Al rigo 15, per “nome del proprietario” si intende nome e cognome del proprietario dell’attività commerciale.

Al rigo 16, per “nome del responsabile” si intende nome e cognome del responsabile dell’attività commerciale.

Ai righi 17 e 18, si intende tenere traccia del nome dell’area geografica, nel modo più generico possibile.

Al rigo 20, per “dati anagrafici” di un dipendente si intende nome, cognome, data di nascita, luogo di nascita, sesso e codice fiscale del dipendente.

Al rigo 22, per “periodo di lavoro” si intende il numero di giorni dalla data di assunzione alla data di licenziamento, oppure alla data corrente.

### Standardizzazione e linearizzazione

Il centro commerciale è costituito da un insieme di edifici nei quali si concentrano numerose attività commerciali.

Le attività commerciali possono essere quelle per la grande distribuzione organizzata, negozi specializzati, cinema e attività di ristorazione.

L’accesso agli edifici che ospitano le diverse imprese commerciali è reso possibile da piccole vie, piazze, gallerie (piazze coperte aperte solo al traffico pedonale).

Tutte le strutture di cui è composto il centro (gli edifici, le piazze, le gallerie, le vie) sono identificate univocamente da una stringa alfanumerica, composta da un carattere identificativo della tipologia di struttura (E per edificio, P per piazza, G per galleria, V per via) e da un numero.

Le varie attività commerciali sono identificate univocamente dal codice della struttura nella quale sono collocate e dal proprio codice numerico.

Per ogni attività commerciale si deve memorizzare il nome della struttura nel quale è collocata ed il piano (nel caso degli edifici).

I negozi sono organizzati per categorie. Alcuni esempi di tipologie di negozi sono l’abbigliamento, l’arredo e l’informatica.

Informazioni di interesse per le attività commerciali sono il nome (o ragione sociale), numero di dipendenti, la partita iva. Se l’attività è in franchising si vuole memorizzare il nome e il cognome del responsabile. Se l’attività non è in franchising si vuole memorizzare il nome e cognome del proprietario.

Anche le attività di ristorazione sono suddivise per tipologia, come i fast-food, i ristoranti tipici di una particolare nazione del mondo o area geografica (italiana o estera).

Il database, infine, tiene memoria di tutti i dipendenti del centro. Per ogni dipendente, si vuole conoscere, insieme ai dati anagrafici, anche la data di assunzione, l’eventuale data di licenziamento, l’attività commerciale presso la quale solo stati assunti e il periodo di lavoro.

### Omonimie e sinonimie

Al rigo 1, per “centro commerciale” si utilizzerà “centro”.

Ai righi 2, 9, 11, 14, 21, per “attività commerciale” si utilizzerà “attività”.

Ai righi 2 e 3, per “negozio specializzato” si utilizzerà “negozio”.

Ai righi 3, 17, per “attività di ristorazione” si utilizzerà “ristorante”.

Al rigo 4, per “impresa commerciale” si utilizzerà “attività”.

Al rigo 13, per “categoria di negozio” e “tipologia di negozio” si utilizzerà “tipo di negozio”.

Al rigo 17, per “tipologia” si utilizzerà “tipo di ristorante”.

### Requisiti trasformati

A questo punto, per facilitare il lavoro nelle fasi successive è possibile riscrivere i requisiti dopo le fasi di linearizzazione, scelta di un corretto livello di astrazione ed eliminazione di omonimie e sinonimie.

|  |
| --- |
| Centro commerciale (requisiti riscritti) |
| Il centro è costituito da un insieme di edifici nei quali si concentrano numerose attività.  Le attività possono essere quelle per la grande distribuzione organizzata, negozi, cinema e ristoranti.  L’accesso agli edifici che ospitano le diverse attività è reso possibile da piccole vie, piazze, gallerie (piazze coperte aperte solo al traffico pedonale).  Tutte le strutture di cui è composto il centro (gli edifici, le piazze, le gallerie, le vie) sono identificate univocamente da una stringa alfanumerica, composta da un carattere identificativo della tipologia di struttura (E per edificio, P per piazza, G per galleria, V per via) e da un numero.  Le varie attività sono identificate univocamente dal codice della struttura nella quale sono collocate e dal proprio codice numerico.  Per ogni attività si deve memorizzare il nome della struttura nel quale è collocata ed il piano (nel caso degli edifici).  I negozi sono organizzati per tipi di negozio. Alcuni esempi di tipi di negozi sono l’abbigliamento, l’arredo e l’informatica.  Informazioni di interesse per le attività sono il nome (o ragione sociale), numero di dipendenti, la partita iva. Se l’attività è in franchising si vuole memorizzare il nome e il cognome del responsabile. Se l’attività non è in franchising si vuole memorizzare il nome e cognome del proprietario.  Anche i ristoranti sono suddivisi per tipo di ristorante, come i fast-food, i ristoranti tipici di una particolare nazione del mondo o area geografica (italiana o estera).  Il database, infine, tiene memoria di tutti i dipendenti del centro. Per ogni dipendente, si vuole conoscere, insieme ai dati anagrafici, anche la data di assunzione, l’eventuale data di licenziamento, l’attività presso la quale solo stati assunti e il periodo di lavoro. |

### Raggruppamento di frasi per concetti

A partire dai requisiti riscritti, è possibile raggruppare le frasi per concetti.

#### Frasi di carattere generale

Si desidera creare un sistema informatico per un centro commerciale.

#### Frasi relative alle strutture

Il centro è costituito da un insieme di edifici nei quali si concentrano numerose attività.

L’accesso agli edifici che ospitano le diverse attività è reso possibile da piccole vie, piazze, gallerie (piazze coperte aperte solo al traffico pedonale).

Tutte le strutture di cui è composto il centro (gli edifici, le piazze, le gallerie, le vie) sono identificate univocamente da una stringa alfanumerica, composta da un carattere identificativo della tipologia di struttura (E per edificio, P per piazza, G per galleria, V per via) e da un numero.

#### Frasi relative alle attività

Le attività possono essere quelle per la grande distribuzione organizzata, negozi, cinema e ristoranti.

Le varie attività sono identificate univocamente dal codice della struttura nella quale sono collocate e dal proprio codice numerico.

Per ogni attività si deve memorizzare il nome della struttura nel quale è collocata ed il piano (nel caso degli edifici).

Informazioni di interesse per le attività sono il nome (o ragione sociale), numero di dipendenti, la partita iva. Se l’attività è in franchising si vuole memorizzare il nome e il cognome del responsabile. Se l’attività non è in franchising si vuole memorizzare il nome e cognome del proprietario.

#### Frasi relative ai negozi

I negozi sono organizzati per tipi di negozio. Alcuni esempi di tipi di negozi sono l’abbigliamento, l’arredo e l’informatica.

#### Frasi relative ai ristoranti

Anche i ristoranti sono suddivisi per tipo di ristorante, come i fast-food, i ristoranti tipici di una particolare nazione del mondo o area geografica (italiana o estera).

#### Frasi relative ai dipendenti

Informazioni di interesse per le attività sono il nome (o ragione sociale), numero di dipendenti, la partita iva (frase ripetuta).

Il database, infine, tiene memoria di tutti i dipendenti del centro. Per ogni dipendente, si vuole conoscere, insieme ai dati anagrafici, anche la data di assunzione, l’eventuale data di licenziamento, l’attività presso la quale solo stati assunti e il periodo di lavoro.

### Glossario dei termini

Nella seguente tabella vengono presentati tutti i concetti ritrovati, collegandoli ad eventuali altre entità e chiarendo eventuali sinonimie ritrovate ai passi precedenti.

|  |  |  |  |
| --- | --- | --- | --- |
| Termine | Descrizione | Sinonimi | Collegamenti |
| Struttura | Contiene attività e può essere di diverso tipo. |  | Attività |
| Attività | Può essere di diverso tipo e si trova in una sola struttura. | Attività commerciale  Impresa commerciale | Struttura  Dipendente |
| Negozio | È un tipo di attività. | Negozio specializzato | Attività  Tipo di negozio |
| Ristorante | È un tipo di attività. | Attività di ristorazione | Attività  Tipo di ristorante |
| Tipo di negozio | Definisce la categoria del negozio. | Categoria  Tipologia di negozio | Negozio |
| Tipo di ristorante | Definisce la categoria del ristorante. | Tipologia di ristorante | Ristorante |
| Dipendente | Persona impiegata per un certo periodo di tempo in un’attività. |  | Attività |

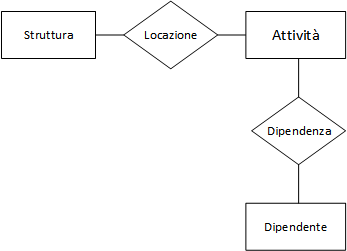
# Progettazione concettuale

Si utilizza la strategie ibrida per progettare lo schema concettuale: a partire da uno schema scheletro contenente, a livello astratto, i concetti principali, si prosegue per raffinamenti successivi o estendendo lo schema con altre entità sviluppate separatamente.

## Schema concettuale

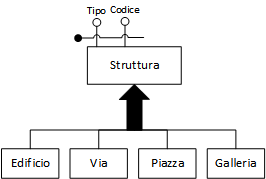
Per la realizzazione dello schema concettuale si è utilizzato Microsoft Visio 2013 (su piattaforma Windows 7).

### Schema scheletro

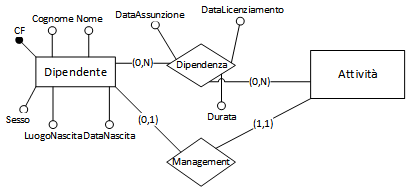


### Esplosione di entità

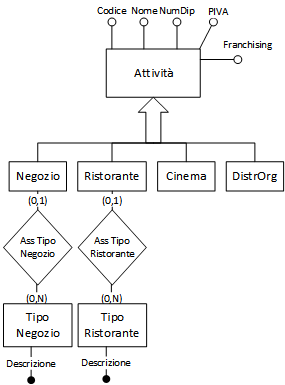
#### Struttura



#### Dipendente

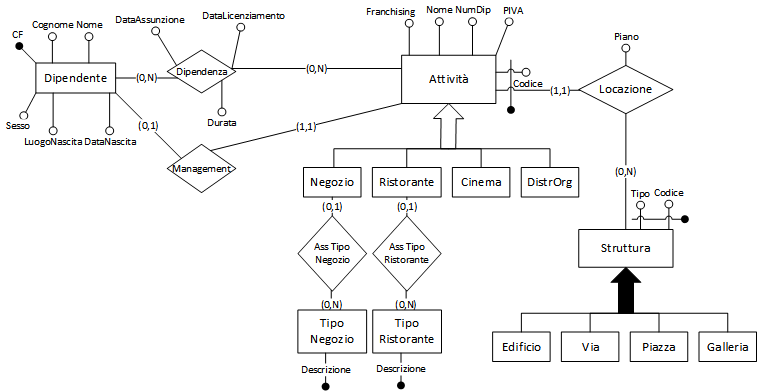


#### Attività



### Schema concettuale finale

Integrando i diversi schemi nello schema scheletro otteniamo uno schema concettuale completo.



## Dizionario dei dati

Per ogni concetto si definisce una descrizione, gli attributi e un identificatore principale.

### Entità

|  |  |  |  |
| --- | --- | --- | --- |
| Entità | Descrizione | Attributi | Identificatore |
| Attività | Può essere di diverso tipo e si trova in una sola struttura. | PIVA Nome  NumDip  Franchising | Codice  Struttura |
| Negozio | È un tipo di attività. |  |  |
| Ristorante | È un tipo di attività. |  |  |
| Cinema | È un tipo di attività. |  |  |
| Distribuzione Organizzata | È un tipo di attività. |  |  |
| Tipo Negozio | Definisce la categoria del negozio. |  | Descrizione |
| Tipo Ristorante | Definisce la categoria del ristorante. |  | Descrizione |
| Struttura | Contiene attività e può essere di diverso tipo. |  | Codice, Tipo |
| Edificio | Tipo di struttura con diversi piani. |  |  |
| Via | Tipo di struttura. |  |  |
| Piazza | Tipo di struttura. |  |  |
| Galleria | Tipo di struttura. |  |  |
| Dipendente | Persona impiegata in qualche attività. | Cognome  Nome  Sesso  LuogoNascita  DataNascita | CF |

### Relazioni

|  |  |  |  |
| --- | --- | --- | --- |
| Relazione | Descrizione | Attributi | Entità |
| Ass Tipo Negozio | Lega un negozio ad un possibile tipo. |  | Negozio (0,1)  Tipo Negozio (0,N) |
| Ass Tipo Ristorante | Lega un ristorante ad un possibile tipo. |  | Ristorante (0,1)  Tipo Ristorante (0,N) |
| Locazione | Collega un’attività ad una specifica struttura. | Piano | Attività (1,1)  Struttura (0,N) |
| Dipendenza | Lega un dipendente a una attività. | DataAssunzione  DataLicenziamento  Durata | Attività (0,N)  Dipendente (0,N) |
| Management | Lega un’attività ad un dipendente che funge da manager. |  | Attività (1,1)  Dipendente (0,1) |

## Regole aziendali

### Regole di vincolo

1. Un dipendente non può avere due dipendenze attive alla stessa data.
2. Ogni volta che viene aggiunto o eliminato un dipendente, si deve aggiornare NumDip dell’attività del dipendente modificato.
3. Periodicamente va aggiornata la durata della dipendenza.
4. Il proprietario o il responsabile di un’attività deve essere un dipendente della stessa.

### Regole di derivazione

1. Il responsabile di una attività si ottiene, se l’attività è in franchising, visitando la relazione Management.
2. Il proprietario di una attività si ottiene, se l’attività non è in franchising, visitando la relazione Management.

# Progettazione logica

La progettazione logica ha come obiettivo quello di tradurre lo schema ER prodotto nella fase precedente in uno schema logico in grado di descrivere gli stessi dati in maniera corretta ed efficiente, prestando particolare attenzione alle prestazioni. Per tale motivo verranno analizzati i dati, le operazioni ed i relativi costi di accesso ad ogni entità e associazione.

## Operazioni

Le operazioni eseguite dal sistema sono:

1. Inserimento di una struttura.
2. Modifica di una struttura.
3. Aggiunta di un’attività pre-esistente ad una struttura.
4. Spostamento di un’attività in una struttura.
5. Assunzione di un dipendente in una attività.
6. Cambio di proprietario di un’attività.
7. Stampa di tutte le attività del centro, incluso il numero dei dipendenti.
8. Stampa della storia delle assunzioni di ogni attività, con data di assunzione, data di licenziamento e durata.
9. Ricerca del proprietario di un’attività.

## Ristrutturazione dello schema E-R

### Tavola dei Volumi

Nella seguente tabella si elenca il volume di ogni classe di entità, in base ad alcune assunzioni presentate successivamente.

|  |  |  |
| --- | --- | --- |
| Concetto | Tipo | Volume |
| Struttura | E | 40 |
| Edificio | E | 10 |
| Via | E | 8 |
| Piazza | E | 7 |
| Galleria | E | 15 |
| Attività | E | 400 |
| Locazione | R | 400 |
| Negozio | E | 250 |
| Ristorante | E | 20 |
| Cinema | E | 5 |
| DistrOrg | E | 10 |
| Tipo Negozio | E | 50 |
| Tipo Ristorante | E | 50 |
| Ass Tipo Negozio | R | 200 |
| Ass Tipo Ristorante | R | 15 |
| Dipendente | E | 15000 |
| Dipendenza | R | 20000 |
| Management | R | 400 |

Sui volumi sono state fatte alcune assunzioni:

* Ogni struttura ha in media 10 attività.
* Ci possono essere altri tipi di attività non specificati dalla gerarchia.
* Quasi tutti i negozi hanno specificato una tipologia di negozio.
* Quasi tutti i ristoranti hanno specificato una tipologia di ristorante.
* Ogni attività ha una media di 20 dipendenze attive in ogni momento.
* Circa 15000 dipendenti hanno transitato da una o più attività del centro nel corso della loro carriera.
* Circa 1 dipendente su 4 ha avuto almeno 2 assunzioni all’interno del centro.

### Tavola delle Operazioni

Per ogni operazione definita in precedenza si definisce la tipologia (interattiva o batch) e la frequenza assunta per l’operazione in un certo periodo di tempo.

|  |  |  |
| --- | --- | --- |
| Op. | Tipo | Frequenza |
| 1 | I | 5/anno |
| 2 | I | 15/mese |
| 3 | I | 5/mese |
| 4 | I | 5/mese |
| 5 | I | 100/mese |
| 6 | I | 3/mese |
| 7 | B | 1/mese |
| 8 | B | 2/anno |
| 9 | I | 10/giorno |

### Tavole degli Accessi

Per ogni operazione si calcolano gli accessi necessari, in lettura e scrittura, ad ogni entità, utilizzando la tavola dei volumi come base per il calcolo.

#### Operazione 1

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Struttura | E | S | 1 |
| Edificio/Via/… | E | S | 1 |

#### Operazione 2

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Struttura | E | S | 1 |
| Edificio/Via/… | E | S | 1 |

#### Operazione 3

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 1 |
| Struttura | E | L | 1 |
| Locazione | R | S | 1 |

#### Operazione 4

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 1 |
| Struttura | E | L | 1 |
| Locazione | R | S | 2 |

Vengono eseguite due attività di scrittura su Locazione, una di cancellazione, l’altra di inserimento.

#### Operazione 5

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Dipendente | E | L | 1 |
| Attività | E | L | 1 |
| Dipendenza | R | S | 1 |
| Attività | E | S | 1 |

Si scrive prima la dipendenza, poi si incrementa il numero dei dipendenti dell’attività.

#### Operazione 6

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 1 |
| Dipendente | E | L | 1 |
| Management | R | S | 2 |

#### Operazione 7

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 400 |

#### Operazione 8

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 400 |
| Dipendenza | R | L | 20000 |
| Dipendente | E | L | 15000 |

#### Operazione 9

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 1 |
| Management | R | L | 1 |
| Dipendente | E | L | 1 |

### Analisi delle ridondanze

Sono presenti due ridondanze nello schema concettuale in ingresso.

#### Durata assunzione

Durata nella relazione Dipendenza è ottenibile tramite l’esecuzione di una semplice differenza fra la data di licenziamento (o quella attuale se la data di licenziamento è nulla) e la data di assunzione.

L’unica operazione che fa uso della durata della dipendenza è l’operazione 8.

Consideriamo il caso in cui la ridondanza rimanga:

* Il costo dell’operazione 8 sarebbe di 35400.
* Ogni ridondanza aggrava la base di dati di 4 byte, quindi si ha uno spreco di spazio di 20000\*4 byte = 80000 byte.

Se la ridondanza viene eliminata:

* Il costo dell’operazione 8 rimane di 35400, in quanto il costo di una sottrazione è irrilevante rispettivamente all’accesso ai blocchi in memoria secondaria.
* Si risparmierebbero 80000 byte.

Pertanto, si decide di rimuovere la ridondanza.

#### Numero dipendenti

Il numero dei dipendenti è ottenibile tramite un semplice conteggio delle dipendenze che hanno DataLicenziamento non impostata.

Le operazioni che fanno uso del numero dei dipendenti sono le operazioni 5 e 7.

Consideriamo il caso in cui la ridondanza rimanga:

* Assumendo che la dimensione dell’attributo NumDip sia di 4 byte, lo spreco di memoria è di 400\*4 byte = 1600 byte.
* Il costo unitario dell’operazione 5 è di 6.
* Il costo unitario dell’operazione 7 è di 400.

Se si considera l’eliminazione della ridondanza, bisogna creare le tavole degli accessi delle operazioni 5 e 7 in sua assenza.

La tavola degli accessi per l’operazione 5 è così definita:

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Dipendente | E | L | 1 |
| Attività | E | L | 1 |
| Dipendenza | R | S | 1 |

La tavola degli accessi per l’operazione 7 è così definita:

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Tipo | Accessi |
| Attività | E | L | 400 |
| Dipendenza | R | L | 8000 |

Se la ridondanza viene eliminata, quindi:

* Si avrebbe un risparmio di memoria di 1600 byte.
* Il costo unitario dell’operazione 5 sarebbe di 4.
* Il costo unitario dell’operazione 7 sarebbe di 8400.

In presenza di ridondanza, quindi, il costo totale sarebbe di .

In assenza di ridondanza, il costo totale sarebbe di .

Poiché il risparmio di memoria è trascurabile, e visto il guadagno di prestazioni ottenuto con la ridondanza, si decide di non eliminare la ridondanza relativa al numero dei dipendenti di un’attività.

### Eliminazione delle gerarchie

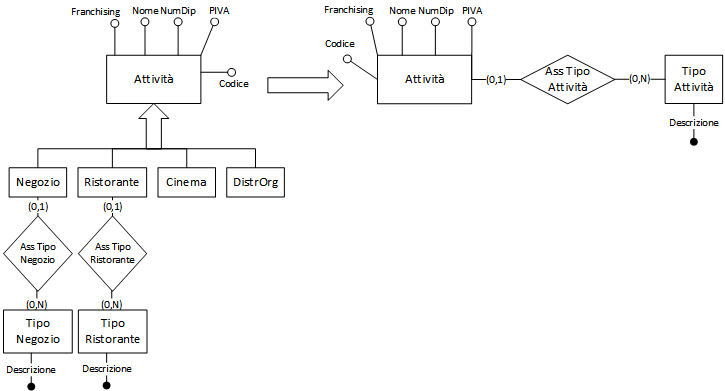
Sono presenti due gerarchie nello schema concettuale.

#### Gerarchia di Attività

Per eliminare la gerarchia di Attività si è scelto di eliminare le entità figlie. Queste, infatti, non partecipano singolarmente a nessuna relazione che non possa essere accorpata nel padre. Poiché ogni attività può avere al massimo un tipo associato, si unificano anche i tipi di attività e si utilizza un’unica relazione per eseguire le associazioni.

Ad esempio, Tipo Attività potrà contenere “Ristorante italiano”, o “Negozio di informatica”, o “Negozio di abbigliamento”, o ancora “Ristorante tipico lucano”.

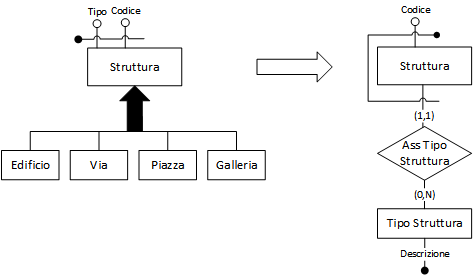
In questo modo sarà possibile inserire qualsiasi nuova attività senza essere forzati a dover scegliere fra le 4 macro-tipologie Negozio, Ristorante, Cinema, DistrOrg.



#### Gerarchia di attività

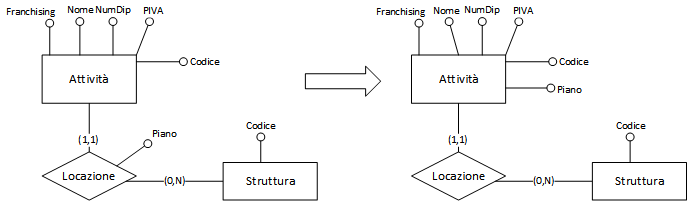
Nella gerarchia di Struttura, le entità figlie non hanno nessun attributo particolare, né partecipano ad alcuna relazione. Pertanto, si può decidere di eliminare la gerarchia, sostituendola con una tipizzazione della Struttura realizzata tramite una relazione.

Si introduce l’entità Tipo Struttura, che contiene una descrizione della tipologia di struttura. Ogni Struttura, in questo modo, dovrà essere di un solo tipo.

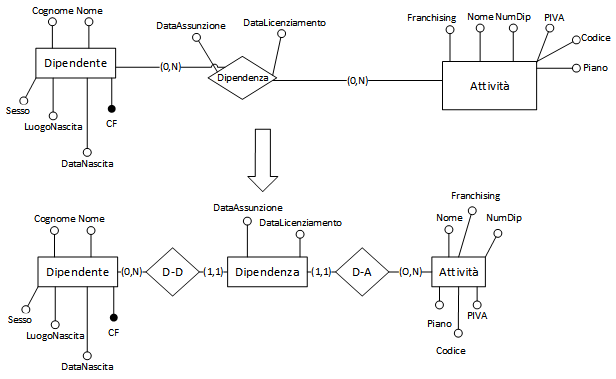


### Partizionamento ed accorpamento di concetti

Locazione è una relazione, ma ha un attributo. Poiché Locazione è obbligatoria, si sposta l’attributo Piano in Struttura, ammettendo valori nulli.



Si sceglie anche di reificare l’associazione Dipendenza, in quanto è un concetto a sé stante e rappresenta una assunzione di un dipendente in un’attività, con due date, una di assunzione e una di licenziamento. Gli attributi dell’associazione vengono spostati sulla nuova entità.



### Scelta degli identificatori principali

Fra tutti gli identificatori principali fin qui ritrovati si sceglie di adottare esclusivamente quelli che possono essere generati dal sistema. Gli altri, essendo passibili di modifica dall’utente, vengono considerati campi univoci (spesso anche con indice univoco), ma non vengono utilizzati come chiave primaria. Nella tabella seguente si elencano tutti gli identificatori delle tabelle.

|  |  |
| --- | --- |
| Concetto | Identificatore |
| Dipendente | ID |
| Dipendenza | ID |
| Attività | IDAttivita |
| Tipo Attività | ID |
| Struttura | IDStruttura |
| Tipo Struttura | ID |

Attività e Struttura partecipano ad alcune relazioni, pertanto un identificatore composto non risulta opportuno.

Per Tipo Struttura il codice viene mantenuto, poiché utile a rappresentare la struttura, ma viene introdotto anche un ID seriale generato automaticamente.

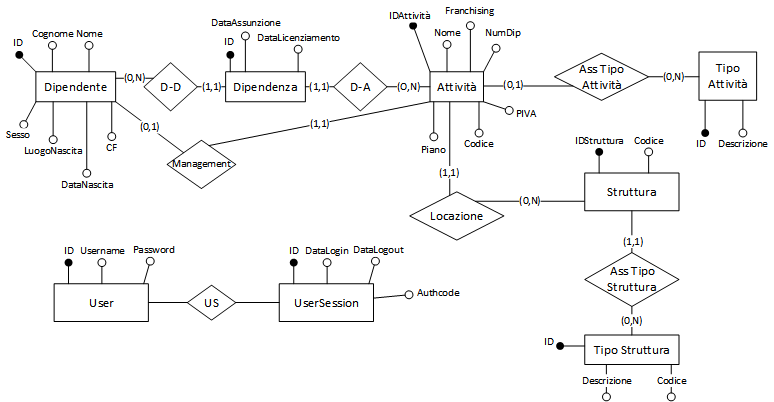
Anche in Tipo Attività e in Dipendenza si genera un ID seriale in maniera automatica, poiché sono entità che possono essere accedute atomicamente.

Per Dipendente l’utilizzo del codice fiscale come chiave primaria viene ritenuto non opportuno, visto che, ai fini applicativi, l’identificatore primario del dipendente sarà inserito nella query string delle richieste al servizio Web.

In generale si è cercato di mantenere l’identificazione degli oggetti la più immediata e compatta possibile.

## Schema ER ristrutturato

A partire dalle analisi fatte lo schema concettuale è stato ristrutturato nel seguente schema ER. Sono state aggiunte, inoltre, due entità e una relazione che servono a gestire la sessione degli utenti che vogliano utilizzare la base di dati dall’apposito applicativo Web.



## Modello logico

Si ottiene il seguente modello logico. Gli identificatori sono sottolineati e gli attributi opzionali sono indicati con un asterisco \*.

* dipendente(**id**, cf, nome, cognome, sesso, luogo\_nascita\*, data\_nascita\*)
* tipo\_struttura(**id**, codice, descrizione)
* struttura(**id\_struttura**, codice, id\_tipo\_struttura)
* tipo\_attivita(**id**, descrizione)
* attivita(**id\_attivita**, nome, num\_dip, piva, codice, franchising, id\_struttura\*, piano\*, id\_tipo\_attivita\*, id\_dipendente\_manager\*)
* dipendenza(**id**, id\_dipendente, id\_attivita, data\_assunzione, data\_licenziamento\*)
* user(**id**, username, password)
* user\_session(**id**, id\_user, date\_login, date\_logout\*, authcode)

I vincoli di chiave esterna sono realizzati dai seguenti attributi.

* struttura.id\_tipo\_struttura è chiave esterna di tipo\_struttura.id
* attivita.id\_struttura è chiave esterna di struttura.id\_struttura
* attivita.id\_tipo\_attivita è chiave esterna di tipo\_attivita.id
* attivita.id\_dipendente\_manager è chiave esterna di dipendente.id
* dipendenza.id\_dipendente è chiave esterna di dipendente.id
* dipendenza.id\_attivita è chiave esterna di attivita.id
* user\_session.id\_user è chiave esterna di user.id

# Progettazione fisica

Si definiscono gli script SQL utilizzati per la creazione del database. In un primo momento si definiscono i tipi e le tabelle alla base dello schema, quindi vengono specificati gli indici, i trigger e le funzioni della base di dati. Gli script di creazione del database sono presenti nel file sql/script.sql.

## Tipi e tabelle

TODO: script di creazione di tipi e tabelle.

## Indici

### Indice b-tree su attivita.nome

TODO: descrizione, motivo, script.

### Indice b-tree su dipendente.cognome e dipendente.nome

TODO: descrizione, motivo, script.

### Indice hash su struttura.codice

TODO: descrizione, motivo, script.

### Indice b-tree su tipo\_attivita.descrizione

TODO: descrizione, motivo, script.

### Indice b-tree su tipo\_struttura.descrizione

TODO: descrizione, motivo, script.

### Indice hash su user.username

TODO: descrizione, motivo, script.

### Indice hash su user\_session.authcode

TODO: descrizione, motivo, script.

## Trigger

### Gestione della ridondanza sul numero dei dipendenti

#### Inserimento

TODO: inserire script.

#### Cancellazione

TODO: Inserire script.

#### Aggiornamento

TODO: Inserire 2 script.

## Funzioni

### Cerca dipendente

La funzione riceve in ingresso una stringa che può essere la combinazione, in qualsiasi ordine, di nome e cognome del dipendente, o di una sua parte. Ad esempio, per ricercare Mario Rossi si potrebbero passare in input le stringhe “mar”, “rossi ma”, “mario r”, ecc. Tale funzione si rivela essenziale per gestire in maniera ottimale l’auto-complete di una eventuale ricerca.

In input vengono anche richiesti un valore massimo di record da restituire ed il valore di record da saltare dall’inizio della ricerca (utile per il paging).

TODO: inserire script.

### Conteggio ricerca dipendenti

Ai fini di creare un sistema di paging sull’applicazione Web, si rende necessario ottenere il numero totale dei dipendenti che rispondono al criterio di ricerca.

### Esistenza di sessione

La funzione riceva in ingresso una stringa che rappresenta un possibile authcode. Se l’authcode viene ritrovato, la funzione restituisce true, altrimenti restituisce false.

TODO: inserire script.

# Servizi Web RESTful

Prima di realizzare l’applicazione Web vera e propria è stato realizzato uno strato di servizi Web di tipo RESTful, basati quindi sul paradigma REST e su un basso accoppiamento fra dati e presentazione. I servizi RESTful sono stati realizzati tramite diverse componenti a più livelli di integrazione:

* **Apache Tomcat** come servlet container;
* **jOOQ** come generatore di entità POJO persistenti su database, per la comunicazione con il database in maniera sicura a compile-time e più efficiente.
* **Jersey** come framework che implementa le specifiche JAX-RS, per la realizzazione dello strato di comunicazione con i client ed esposizione dell’interfaccia del servizio Web.

Il progetto del servizio Web è stato realizzato con Eclipse Juno e JRE 1.6. IL DBMS utilizzato è PostgreSQL 9.2.

## Autenticazione e sicurezza

TODO: aggiungere gestione dell’autenticazione.

## Cross Origin Resource Sharing

Al fine di prevedere l’utilizzo dello strato di servizi da qualsiasi server (e da qualsiasi porta), è stato necessario abilitare il CORS (Cross Origin Resource Sharing), che permette a qualsiasi server di richiedere e modificare risorse esposte dal RESTful Service.

La funzionalità è abilitata inserendo, ad ogni richiesta ricevuta da un client, l’header Access-Control-Allow-Origin impostato su \*, in modo tale da permettere richieste da qualsiasi origine (combinazione di indirizzo e porta). Della gestione CORS si occupa la classe CorsResponseFilter.

## Risorse esposte

Di seguito si elencano le risorse esposte dal servizio Web, con relativi metodi accettati.

TODO: aggiungere risorse ed azioni supportate.

# Web Application

Per realizzare l’applicazione Web sono stati utilizzati diversi strumenti e tecnologie:

* Come base dell’applicazione è stato utilizzato un emergente framework MVC lato client, **AngularJS** (di Google). Fortemente basato sul concetto di Dependency Injection, esso permette di utilizzare moduli di codice “al volo”, definiti in un qualsiasi altro punto dell’applicazione. Possiede un linguaggio di templating, funzioni automatiche di two-way data binding, moduli di rete, sistemi per la definizione di componenti HTML riutilizzabili, filtri sui dati, direttive HTML, servizi e factories.
* Per la parte grafica si è utilizzato il pacchetto grafico **Bootstrap** (di Twitter), che mette a disposizione stili CSS predefiniti e plugin JavaScript che favoriscono una presentazione di tipo “responsive”.
* **bootstrap-datepicker** è un plugin per Bootstrap che permette di utilizzare un datepicker su un tag input.
* **angular-resource** è una parte opzionale di AngularJS che espone un service per la definizione e l’utilizzo di risorse REST in maniera immediata, con pieno supporto al CORS e al caching automatico (opzionale).
* **angular-cookies** è un altro modulo opzionale di AngularJS che permette di gestire i cookie nello stesso modo su tutti i browser, eliminando le incongruenze e fornendo un accesso top-level ai cookies tramite un oggetto associativo
* **angular-ui** è un pacchetto aggiuntivo di AngularJS che fornisce ulteriori direttive, filtri e componenti.
* **angular-bootstrap** e **angular-strap** forniscono direttive specifiche per i componenti di Twitter Bootstrap in AngularJS.
* **moment** è una libreria per la gestione delle date in Javascript, che fornisce wrapper di accesso alla data equivalenti per tutti i browser.
* **yeoman** è un tool per la gestione del progetto a livello di dipendenze fra pacchetti, building e testing.
* **jQuery** viene usato (opzionalmente nella versione lite) per la manipolazione del DOM da AngularJS.
* L’IDE scelto per lo sviluppo delle diverse componenti dell’applicazione Web è **JetBrains WebStorm** 5.0.4.

## Componenti MVC

### Router

Il Router utilizzato è quello integrato in AngularJS, $routeProvider, che permette di far corrispondere, a determinati template URL, un Controller ed una View, che vengono automaticamente iniettati nel contesto applicativo e nel DOM.

Seguendo il paradigma REST anche per l’applicazione Web, ogni URL identifica una particolare risorsa ed azione relativa.

TODO: aggiungere risorse ed azioni supportate.

### Controller

In conformità al paradigma MVC, ogni Controller è associato ad una View e viene iniettato automaticamente in corrispondenza di una View che lo dichiara o di una template URL risolta dal $routeProvider.

Ogni Controller ha un proprio oggetto $scope, che delimita un “confine” entro il quale i componenti della View possono controllare e modificare gli oggetti. I Controller, comunque, possono essere innestati uno dentro l’altro se le View sono annidate fra di loro. Esiste anche un super-contesto a livello padre, chiamato $rootScope.

TODO: aggiungere Controller ed azioni compiute da essi.

### View

Le View sono, banalmente, pagine parziali che possono essere iniettate in qualsiasi punto dell’applicazione e che sono fortemente accoppiate ad un Controller che gestisce i dati che esse presentano.

TODO: aggiungere View e relativi Controller.

## Comunicazione con il servizio RESTful

La comunicazione con il servizio RESTful è stata realizzata facendo affidamento al modulo ngResource: si è definita una Factory per ogni risorsa sul servizio (in /scripts/services/), che espone e media la comunicazione fra i Controller ed il modulo $resource di AngularJS.

# Progettazione del Data Warehouse

Si può supporre di voler integrare i dati amministrativi del centro commerciale, gestiti dall’applicativo Web realizzato, con altri sistemi di gestione della contabilità e degli incassi, magari pre-esistenti, al fine di fornire uno strumento di analisi per gli analisti dei dati e per gli analisti di business.

A causa dell’eterogeneità delle tipologie di attività, e quindi di prodotti venduti e servizi offerti, si potrebbe immaginare di richiedere a tutte le entità del centro commerciale di fornire periodicamente, tramite opportuni sistemi di acquisizione dati, informazioni generiche relative a:

* Incasso
* Attività che ha generato l'incasso
* Data
* Dipendente che ha portato a termine l’acquisto (dove applicabile)

Il sistema di acquisizione a partire dalle informazioni ricevute le integra con i dati presenti nel database di gestione del centro commerciale già realizzato, andando ad aggiungere le seguenti informazioni:

* Tipologia dell’attività
* Struttura nella quale l’attività si trova al momento dell’incasso
* Eventuale manager o proprietario dell’attività, detto “responsabile”.

## Progettazione concettuale

Andiamo a definire lo schema concettuale del Data Warehouse che si vuole creare.

Alla base del DW vi è un cubo multidimensionale così definito:

* Fatto: incasso
* Misure: ammontare dell’incasso
* Dimensioni:
  + Attività
  + Data
  + Dipendente
  + Struttura
  + Responsabile

La tabella dei fatti rappresenta la struttura centrale nello schema a stella, dove le entità connesse da opportune relazioni rappresentano le dimensioni di analisi:

* Attività, contiene le informazioni sulle attività che generano gli incassi.
* Data, contiene le date relative agli incassi, con possibilità di realizzare gerarchie su giorno, mese, trimestre, anno.
* Dipendente, contiene le informazioni sui dipendenti che hanno concluso una vendita o realizzato un servizio, dove applicabile.
* Struttura, contiene la struttura in cui si trova l’attività al momento dell’incasso. Una attività, infatti, potrebbe cambiare struttura.
* Responsabile, contiene il manager o il proprietario (informazioni opzionali) dell’attività al momento dell’incasso.

Sarebbe stato possibile introdurre un’ulteriore collegamento fra Attività e tipo di attività. Tuttavia si è scelto di optare verso una soluzione che garantisca migliori prestazioni a discapito del piccolo spreco di spazio.

## Progettazione logica

Lo schema a stella/a costellazione è stato tradotto in un opportuno schema logico relazionale in modo tale da poterlo implementare tramite il DBMS di destinazione scelto, PostgreSQL 9.2.

## Implementazione

Si definiscono gli script SQL utilizzati per la creazione del DW, presenti nel file dw/script.sql. È stato utilizzato il pacchetto open source Mondrian per la definizione e l’utilizzo del cubo multidimensionale realizzato tramite il DW qui presentato.

Il software Schema Workbench è stato utilizzato per creare lo schema multidimensionale a partire dal DW di tipo relazionale. È stato creato un file XML che definisce il cubo, la tabella dei fatti con misure, le dimensioni e le relative gerarchie. È possibile ritrovare l’XML al percorso dw/cube.xml.

È possibile esplorare ed analizzare il cubo tramite la pagina TODO: inserire pagina, realizzata tramite il plugin TODO: inserire eventuale plugin.