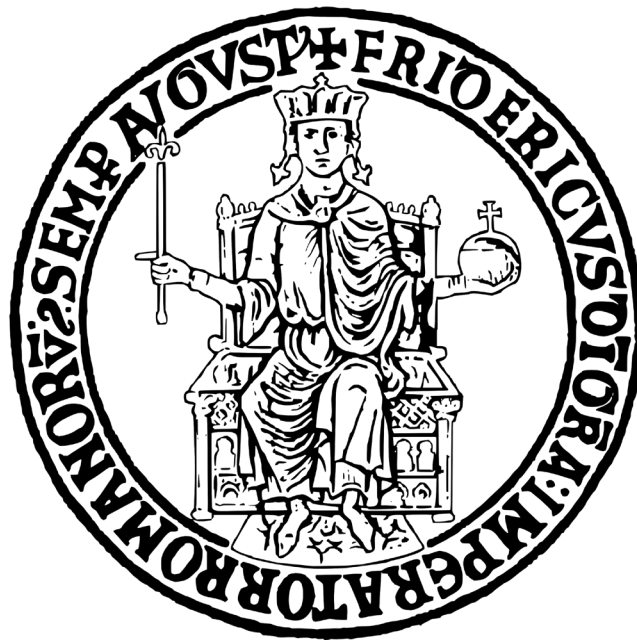


*Università degli studi di Napoli Federico II*  
*Scuola Politecnica e delle Scienze di Base*

*Dipartimento di Ingegneria Elettrica*  
*e Tecnologie dell'Informazione*



*Anno accademico 2020/2021*

*Progettazione e sviluppo di una base di dati  
relazionale per la gestione di progetti.*

*Esposito Sara*

*Trinchillo Giusi Ornella*

*Beniamino Gentile*

*N86003374*

*N86003950*

*N86003004*

# *Indice*

<b>1/</b>	<b><i>Descrizione del progetto.....</i></b>	<b><i>pag.3</i></b>
1.1	<i>Analisi progetto.....</i>	<i>pag.3</i>
<b>2/</b>	<b><i>Progettazione concettuale.....</i></b>	<b><i>pag.4</i></b>
2.1	<i>Class Diagram.....</i>	<i>pag.4</i>
2.2	<i>Alcune precisazioni sul class diagra.....</i>	<i>pag.5</i>
2.3	<i>Descrizione class diagra.....</i>	<i>pag.5</i>
2.4	<i>Ristrutturazione del class diagram.....</i>	<i>pag.5</i>
2.5	<i>Chiavi primari.....</i>	<i>pag.6</i>
2.6	<i>Attributi multipli.....</i>	<i>pag.6</i>
2.7	<i>Attributi derivati.....</i>	<i>pag.6</i>
2.8	<i>Attributi strutturati.....</i>	<i>pag.6</i>
2.9	<i>Gerarchie di specializzazione.....</i>	<i>pag.6</i>
2.10	<i>Class Diagram Revisionato.....</i>	<i>pag.7</i>
<b>3/</b>	<b><i>Dizionari dei dati.....</i></b>	<b><i>pag.8</i></b>
3.1	<i>Dizionario delle classi.....</i>	<i>pag.8</i>
3.2	<i>Dizionario delle associazioni.....</i>	<i>pag.11</i>
3.3	<i>Dizionario dei vincoli.....</i>	<i>pag.14</i>
<b>4/</b>	<b><i>Progettazione Logica... ..</i></b>	<b><i>pag.16</i></b>
4.1	<i>Schema Logico.....</i>	<i>pag.16</i>

# *Capitolo 1*

## *Descrizione del progetto*

### *1.1 Analisi del problema*

*Si progetti ed implementi una base di dati relazionale, ovvero uno schema concettuale di dati strutturati, che possa essere d'aiuto alla gestione e memorizzazione di progetti realizzati da un'azienda.*

*Bisogna tener traccia dei dipendenti della suddetta azienda ricordando che ognuno di essi ha un ruolo, in particolare per ogni progetto ci sarà un unico project manager.*

*Il progetto avrà una specifica tipologia, che può essere: "Ricerca di base", "Ricerca Industriale", "Ricerca sperimentale", "Sviluppo Sperimentale", ed uno o più ambiti, tra cui: Economia, Medicina, Informatica.*

*Per ogni dipendente ci sarà un identificativo reale: il codice fiscale.*

*Al fine di collaborare alla realizzazione del progetto ogni gruppo potrà partecipare a dei meeting telematici o fisici.*

*Il project manager deve poter accedere ai registri delle presenze dei propri colleghi, alle loro skill e al loro salario al fine dell'organizzazione dei gruppi e della valutazione aziendale di ogni singolo membro.*

# Capitolo 2

## Progettazione concettuale

### 2.1 Class Diagram

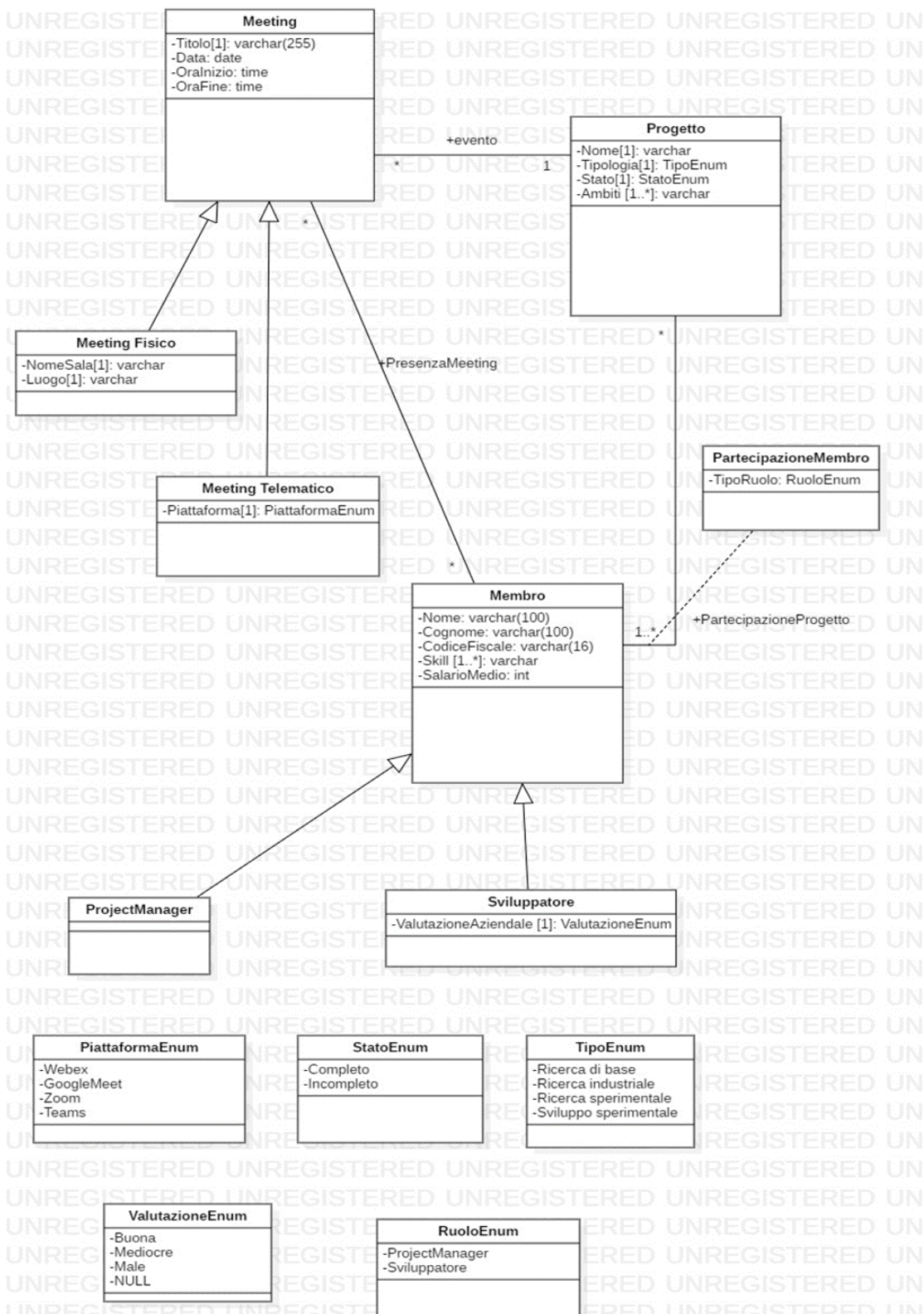


Fig.1

## 2.2 Alcune precisazioni sul class diagram

Al fine di semplificare la lettura si è scelto di:

- Esplicitare le enumerazioni in basso.
- Non impostare la lunghezza degli attributi di tipo VARCHAR a meno che non sia di fondamentale importanza, come nel caso del codice fiscale.
- Le cardinalità delle associazioni indicate con \* specificano che quella classe può partecipare da un minimo di zero a un massimo di molti, ossia (0,\*).
- Dare un solo nome alle associazioni invece di due (uno per direzione).

## 2.3 Descrizione class diagram

Dal class diagram (Fig.1) si può osservare che sono state create tre classi principali per le entità **membro**, **progetto**, **meeting**, due classi di specializzazione di meeting: **meeting telematico** e **meeting fisico** e due classi di specializzazione di membro: **ProjectManager** e **Sviluppatore**.

Tra meeting e membro è stata inserita l'associazione **PresenzaMeeting**, questa associazione è chiamata così perché esplicita:

- a quanti meeting un membro ha preso parte con cardinalità (0,\*) perché se un membro lavora a un progetto da solo (cosa che è possibile fare secondo l'associazione **PartecipazioneProgetto**) non ha bisogno di partecipare a meeting.
- quanti membri partecipano ai meeting con cardinalità (1,\*)

L'associazione **PartecipazioneProgetto** ha il compito di collegare la classe membro e la classe progetto con cardinalità:

- (1,\*) perché un progetto può essere formato da un solo membro.
- (\*) perché un membro può partecipare a nessuno (essere in vacanza o periodo di stop) o più progetti.

## 2.4 Ristrutturazione del class diagram

Al fine di rendere il class diagram idoneo alla traduzione in schemi relazionali e di migliorare l'efficienza dell'implementazione si procede alla ristrutturazione dello stesso.

È necessario analizzare le chiavi primarie e fare in modo che siano univoche e più sintetiche possibili. Anche gli attributi derivati, se possibile, devono essere omessi.

Al termine del procedimento di ristrutturazione il class diagram (Fig.2) non conterrà attributi strutturati, attributi multipli, gerarchie di specializzazione e associazioni.

## 2.5 Chiavi Primarie

Nel class diagram ristrutturato si è deciso, al fine di agevolare l'identificazione, di aggiungere un attributo destinato a essere la **chiave primaria** della classe qualora la chiave primaria fosse composta.

Ad esempio per le classi "progetto" e "meeting" sono state aggiunte le chiavi primarie "CodMeet" e "CodProgetto".

## 2.6 Attributi Multipli

È necessario gestire gli **attributi multipli** in quanto questi prevedono la possibilità di contenere più valori. A causa dell'impossibilità di stimare il numero di diversi attributi associati creeremo una nuova classe contenente l'attributo.

Per esempio la classe "Membro" contiene l'attributo "Skill", quest'ultimo è un attributo multiplo in quanto a un membro potrebbero essere associate più skills. Per evitare di avere un attributo di cui non si sa con precisione a che valore riferirsi, si crea una classe "**Skill**" apposita.

Un altro attributo multiplo proviene dalla classe "Progetto": ambito. Un progetto, infatti, può trattare più ambiti. Anche questa volta creiamo una classe per "**Ambito**". Facendo ciò però causeremo un'associazione molti a molti, poiché un progetto può includere più ambiti e un ambito è trattato da più progetti.

## 2.7 Attributi Derivati

Nel class diagram mostrato non sono presenti attributi derivati.

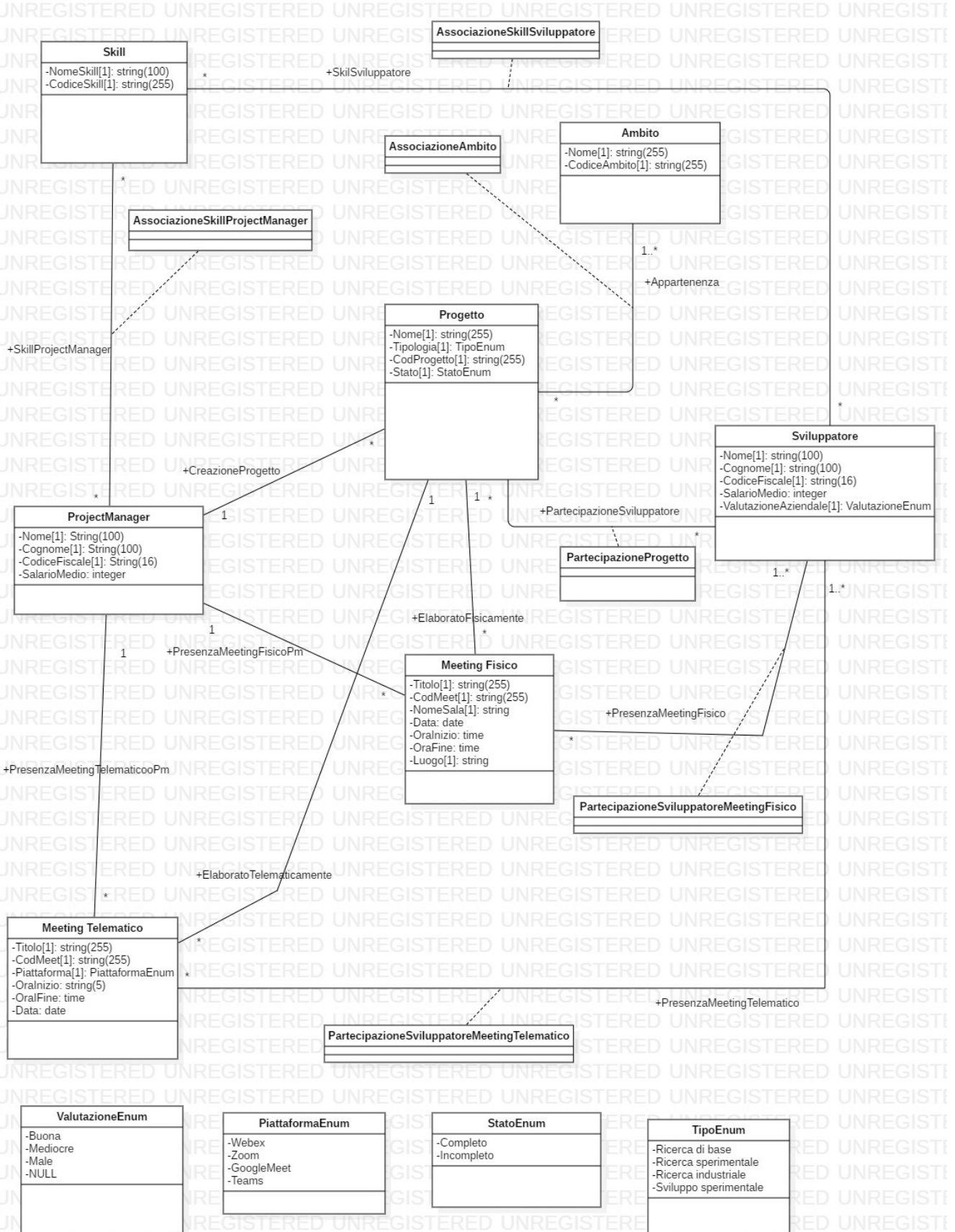
## 2.8 Attributi Strutturati

Nel class diagram mostrato non sono presenti attributi strutturati.

## 2.9 Gerarchie di Specializzazione

Nel primo class diagram si hanno due tipi di specializzazioni. Esistono più modi per eliminare le specializzazioni, si è scelto di accorpare la classe padre nelle classi figlie per la generalizzazione "Meeting", infatti non troviamo più la classe "Meeting" ma i suoi attributi sono stati trasferiti nelle due classi figlie: Meeting Telematico e Meeting Fisico. Lo stesso è avvenuto per la classe Membro e le sue specializzazioni, sono diventate due classi: Sviluppatore e ProjectManager.

## 2.10 Class Diagram Revisionato



*Fig. 2*

## Capitolo 3

### Dizionario dei dati

#### 3.1 Dizionario delle classi

<i>Classe</i>	<i>Descrizione</i>	<i>Attributo</i>
<i>ProjectManager</i>	<i>È la classe che descrive un impiegato di un'azienda nel ruolo di Project Manager. Ha il compito dell'organizzazione aziendale.</i>	<i>CodiceFiscale:</i> (String(16)) Chiave primaria che identifica univocamente i project manager.  <i>Nome:</i> (String(100)) nome associato al project manager.  <i>Cognome:</i> (String(100)) cognome associato al project manager.  <i>SalariaMedio:</i> (integer) il salario medio che riceve il project manager.
<i>Sviluppatore</i>	<i>È la classe che descrive un impiegato di un'azienda nel ruolo di sviluppatore. Ha il compito di svolgere le mansioni che gli vengono affidate.</i>	<i>CodiceFiscale:</i> (String(16)) Chiave primaria che identifica univocamente uno sviluppatore.  <i>Nome:</i> (String(100)) nome associato allo sviluppatore.  <i>Cognome:</i> (String(100)) cognome associato allo sviluppatore.  <i>SalariaMedio:</i> (integer) il salario medio che lo sviluppatore riceve.  <i>ValutazioneAziendale:</i> (ValutazioneEnum) la valutazione che un project manager associa ad uno sviluppatore.



<b>Progetto</b>	<i>È la classe che descrive un progetto realizzato o da realizzare dai membri di un'azienda.</i>	<p><b>CodProgetto:</b> (String(255)) Chiave primaria che identifica univocamente un progetto.</p> <p><b>Nome:</b> (String(255)) nome dato al progetto.</p> <p><b>Tipologia:</b> (TipoEnum) indica il tipo di progetto.</p> <p><b>Stato:</b> (StatoEnum) indica se il progetto è stato portato a termine o meno.</p>
<b>Meeting Fisico</b>	<i>È la classe che identifica un meeting tra i membri di un'azienda, di tipo fisico.</i>	<p><b>CodMeet:</b> (String(255)) Chiave primaria che identifica univocamente un meeting fisico.</p> <p><b>Titolo:</b> (String(255)) nome assegnato ad un meeting fisico come titolo.</p> <p><b>NomeSala:</b> (String) nome della sala dove si svolge il meeting.</p> <p><b>Data:</b> (Date) il giorno, il mese e l'anno in cui si svolge il meeting.</p> <p><b>OraInizio:</b> (Time) l'orario di inizio del meeting.</p> <p><b>OraFine:</b> (Time) l'ora di fine del meeting.</p> <p><b>Luogo:</b> (String) il luogo in cui si svolge il meeting.</p>
<b>Meeting Telematico</b>	<i>È la classe che identifica un meeting tra i membri di un'azienda, di tipo telematico.</i>	<p><b>CodMeet:</b> (String(255)) Chiave primaria che identifica univocamente un meeting telematico.</p> <p><b>Titolo:</b> (String(255)) nome</p>

		<p>assegnato ad un meeting telematico come titolo.</p> <p><b>Piattaforma:</b> (PiattaformaEnum) il sito/app in cui si svolge il meeting telematico.</p> <p><b>Data:</b> (Date) il giorno, il mese e l'anno in cui si svolge il meeting.</p> <p><b>OraInizio:</b> (Time) l'orario di inizio del meeting.</p> <p><b>OraFine:</b> (Time) l'ora di fine del meeting.</p>
<b>Skill</b>	È la classe che identifica le abilità di un membro (Project manager o sviluppatore).	<p><b>CodiceSkill:</b> (String (255)) Chiave primaria che identifica univocamente una skill.</p> <p><b>NomeSkill:</b> (String(100)) il nome di una skill.</p>
<b>Ambito</b>	È la classe che identifica l'ambito di un progetto.	<p><b>CodiceAmbito:</b> (String (255)) Chiave primaria che identifica univocamente un ambito.</p> <p><b>NomeAmbito:</b> (String(255)) il nome di un ambito.</p>

## 3.2 Dizionario delle associazioni

<i>Associazioni</i>	<i>Descrizione</i>	<i>Classi coinvolte</i>
<i>CreazioneProgetto</i>	<i>Esprime la possibilità per un project manager di creare un progetto.</i>	<p><i>Progetto[*]</i> ruolo <b>è creato</b>: indica il progetto che viene creato dal project manager.</p> <p><i>ProjectManager[1]</i> ruolo <b>crea</b>: indica il project manager che crea il progetto.</p>
<i>Partecipazione ProgettoSviluppatore</i>	<i>Esprime la possibilità per uno sviluppatore di partecipare a un progetto.</i>	<p><i>Sviluppatore[*]</i> ruolo <b>lavora ad</b>: indica lo sviluppatore che lavora ad un progetto.</p> <p><i>Progetto[*]</i> ruolo <b>viene realizzato</b>: indica il progetto che viene realizzato da uno o più sviluppatori.</p>
<i>PresenzaMeetingTelematico</i>	<i>Esprime la possibilità da parte di uno sviluppatore di partecipare a un meeting in via telematica.</i>	<p><i>Sviluppatore[1..*]</i> ruolo <b>partecipa</b>: indica lo sviluppatore che presenzia a un meeting telematico.</p> <p><i>MeetingTelematico[*]</i> ruolo <b>ha partecipanti</b>: indica il meeting telematico che ha come partecipanti degli sviluppatori.</p>
<i>PresenzaMeetingFisico</i>	<i>Esprime la possibilità da parte di uno sviluppatore di partecipare ad un meeting fisico.</i>	<p><i>Sviluppatore[1..*]</i> ruolo <b>partecipa</b>: indica lo sviluppatore che presenzia a un meeting fisico.</p> <p><i>MeetingFisico[*]</i> ruolo <b>ha partecipanti</b>: indica il</p>

		meeting fisico che ha come partecipanti degli sviluppatori.
<i>PresenzaMeetingFisicoPm</i>	Esprime la possibilità da parte di un project manager di partecipare ad un meeting in via telematica.	<p><b>ProjectManager[1]</b> ruolo <b>organizzatore</b>: indica il project manager che organizza un meeting fisico.</p> <p><b>MeetingFisico[*]</b> ruolo è <b>organizzato</b>: indica il meeting fisico che viene organizzato da un solo project manager.</p>
<i>PresenzaMeetingTelematicoPm</i>	Esprime la possibilità da parte di un project manager di partecipare ad un meeting fisico.	<p><b>ProjectManager[1]</b> ruolo <b>organizzatore</b>: indica il project manager che organizza un meeting telematico.</p> <p><b>MeetingTelematico[*]</b> ruolo è <b>organizzato</b>: indica il meeting telematico che viene organizzato da un solo project manager.</p>
<i>ElaboratoTelematicamente</i>	Esprime la possibilità di elaborare un progetto durante un meeting telematico.	<p><b>MeetingTelematico[*]</b> ruolo <b>permette elaborazione</b>: indica il meeting telematico dove viene elaborato un progetto.</p> <p><b>Progetto[1]</b> ruolo <b>viene elaborato</b>: indica il progetto che viene elaborato durante un meeting telematico.</p>

<b>ElaboratoFisicamente</b>	Esprime la possibilità di elaborare un progetto durante un meeting fisicamente.	<p><b>MeetingFisico[*]</b> ruolo <b>permette elaborazione</b>: indica il meeting fisico dove viene elaborato un progetto.</p> <p><b>Progetto[1]</b> ruolo <b>viene elaborato</b>: indica il progetto che viene elaborato durante un meeting fisico.</p>
<b>SkillSviluppatore</b>	Esprime le abilità degli sviluppatori.	<p><b>Sviluppatore[*]</b> ruolo <b>ha</b>: indica gli sviluppatori che possiedono una determinata skill.</p> <p><b>Skill[*]</b> ruolo <b>è posseduta</b>: indica le skill sono possedute da uno sviluppatore.</p>
<b>SkillProjectManager</b>	Esprime le abilità dei project manager.	<p><b>ProjectManager[*]</b> ruolo <b>ha</b>: indica i project manager che possiedono una determinata skill.</p> <p><b>Skill: [*]</b> ruolo <b>è posseduta</b>: indica le skill che sono possedute da un project manager.</p>
<b>Appartenenza</b>	Esprime l'appartenenza di un progetto a uno o più ambiti.	<p><b>Ambito[*]</b> ruolo <b>appartiene</b>: indica l'ambito a cui appartiene un progetto.</p> <p><b>Progetto[*]</b> ruolo <b>ha appartenenza</b>: indica il progetto che ha appartenenza a un determinato ambito.</p>

### 3.3 Dizionario dei vincoli

<b><i>Vincoli</i></b>	<b><i>Descrizione</i></b>
<b><i>Legit CodiceFiscale</i></b>	<i>I codici fiscali degli sviluppatori e dei project manager devono essere di forma legittima, ovvero essere di 16 caratteri: i primi sei caratteri sono lettere da 'A-Z', i successivi due sono numeri dallo '0-9', ancora una lettera, due numeri, una lettera, tre numeri e una lettera.</i>
<b><i>Legit valutazione</i></b>	<i>La valutazione assegnata agli sviluppatori può essere 'Buona', 'Mediocre' o 'Male', oppure di default sarà NULL.</i>
<b><i>Legit tipologia</i></b>	<i>La tipologia di un progetto può essere 'Ricerca di base', 'Ricerca industriale', 'Ricerca sperimentale' o 'Sviluppo sperimentale'.</i>
<b><i>Legit stato</i></b>	<i>Lo stato di un progetto può essere 'Completo' o 'Incompleto'. Un progetto nasce come 'Incompleto' ma nel corso del tempo può cambiare il suo stato.</i>
<b><i>Distinct Sviluppatore/ProjectManager in un progetto</i></b>	<i>In ogni progetto può partecipare al più un membro con lo stesso codice fiscale. Questo è un vincolo necessario nel momento in cui un membro può essere sia project manager in un progetto sia sviluppatore in un altro progetto.</i>
<b><i>Distinct Nome Skill</i></b>	<i>Le skills non possono avere nomi uguali. Un nome può essere associato al più a un'unica skill.</i>
<b><i>Distinct Codice Skill</i></b>	<i>Le skills non possono avere codici uguali. Il codice di una skill può essere associato al più a un'unica skill.</i>
<b><i>Distinct Codice Ambito</i></b>	<i>Gli ambiti non possono avere codici uguali. Il codice di un ambito può essere associato al più a un unico ambito.</i>
<b><i>Distinct Codice Progetto</i></b>	<i>I progetti non possono avere codici uguali. Il codice di un progetto può essere associato al più a un unico progetto.</i>

<i><b>Distinct Codice e Nome Meeting fisico</b></i>	<i>I meeting fisici non possono avere codici o nomi uguali. Il codice o il nome di un meeting fisico può essere associato al più ad un unico meeting fisico.</i>
<i><b>Distinct Codice e Nome Meeting telematico</b></i>	<i>I meeting telematici non possono avere codici o nomi uguali. Il codice o il nome di un meeting telematico può essere associato al più ad un unico meeting telematico.</i>
<i><b>Legit Time Meeting</b></i>	<i>Non è possibile che l'ora di inizio di un meeting (telematico o fisico) sia successiva all'orario di fine.</i>

# Capitolo 4

## Progettazione Logica

In questo capitolo sarà considerata la fase successiva della progettazione della base di dati traducendo lo schema concettuale in uno schema logico, dipendente dal tipo di struttura dei dati prescelto cioè nel nostro caso quello relazionale.

Negli schemi relazionali che seguiranno le chiavi primarie sono indicate con una singola sottolineatura mentre le chiavi esterne con una doppia sottolineatura.

### 4.1 Schema Logico

**ProjectManager** (Nome, Cognome, CodiceFiscale, SalarioMedio)

**Sviluppatore** (Nome, Cognome, CodiceFiscale, SalarioMedio, ValutazioneAziendale)

**MeetingFisico** (Titolo, CodMeet, NomeSala, Data, OraInizio, OraFine, Luogo, codProgetto, codProjectManager)

*codProgetto -> Progetto.codProgetto*

*codProjectManager -> ProjectManager.CodiceFiscale*

**PartecipazioniSviluppatoriMeetingFisico** (codFiscale, codMeeting)

*codFiscale -> Sviluppatore.CodFiscale*

*codMeeting -> MeetingFisico.codMeet*



**MeetingTelematico** (Titolo, CodMeet, Piattaforma, Data, OraInizio, OraFine, codProgetto, codProjectManager)

*codFiscale -> Sviluppatore.CodFiscale*

*codMeeting -> MeetingTelematico.codMeet*

**Skill** (NomeSkill, CodiceSkill)

**AssociazioneSkillsProjectManager** (codFiscale, codSkills)

*codFiscale -> ProjectManager.CodiceFiscale*

*codSkills -> Skill.codiceSkill*

**AssociazioneSkillsSviluppatore** (codFiscale, codSkills)

*codFiscale -> Sviluppatore.CodFiscale*

*codSkills -> Skill.codiceSkill*

**Progetto** (Nome, Tipologia, CodProgetto, Stato, CodFiscale)

*CodFiscale -> ProjectManager.CodiceFiscale*

**PartecipazioniProgetto** (codFiscale, codProgetto)

*codFiscale -> Sviluppatore.CodFiscale*

*codProgetto -> Progetto.codProgetto*

**Ambito** (Nome, CodiceAmbito)

**AssociazioneAmbito** (CodProgetto, codAmbito)

*codProgetto -> Progetto.codProgetto*

*codAmbito -> Ambito.codiceAmbito*