

**UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II  
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE  
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE  
DELL'INFORMAZIONE**



CORSO DI LAUREA IN INFORMATICA  
INSEGNAMENTO DI BASI DI DATI E SISTEMI INFORMATIVI I  
ANNO ACCADEMICO 2020/2021

Progettazione e sviluppo di un database  
relazionale per la descrizione di informazioni  
rilevanti ai fini della gestione di uno o più  
Cantieri.

*Autore :*

Giorgio Formicola  
MATRICOLA : N86002220  
[giorf.formicola@studenti.unina.it](mailto:giorf.formicola@studenti.unina.it)

### *Docenti :*

Prof. Adriano Peron  
Prof. Silvio Barra



# INDICE

---

1   Descrizione del progetto .....	5
1.1 : Analisi del problema .....	5
2   Progettazione concettuale .....	7
2.1 : Introduzione .....	7
2.2 : Class Diagram .....	7
2.3 : Ristrutturazione Class Diagram .....	9
2.3.1 : Analisi gerarchie specializzazioni .....	9
2.3.2 : Analisi delle ridondanze e attributi calcolabili .....	9
2.3.3 : Analisi e rimozione di attributi multipli .....	9
2.3.4 : Analisi degli identificativi .....	10
2.3.5 : Analisi delle associazioni molti a molti .....	10
2.4 : Dizionari .....	12
2.4.1 : Dizionario delle classi .....	12
2.4.2 : Dizionario delle associazioni .....	15
2.4.3 : Dizionario dei vincoli .....	17
3   Progettazione logica .....	19
3.1 : Introduzione .....	19
3.2 : Schema logico .....	19
4   Progettazione fisica .....	21
4.1 : Introduzione .....	21
4.2 : Definizione delle tabelle .....	21
4.2.1 : Definizione tabella amministratore .....	21
4.2.2 : Definizione tabella capo cantiere .....	22
4.2.3 : Definizione tabella operaio .....	22

4.2.4 : Definizione tabella cantiere .....	23
4.2.5 : Definizione tabella sensore .....	23
4.2.6 : Definizione tabella ruolo .....	24
4.2.7 : Definizione tabella area cantiere .....	24
4.2.8 : Definizione tabella montaggio sensore .....	25
4.2.9 : Definizione tabella report sensore .....	25
4.3 : Definizione procedure trigger e regexp .....	26
4.3.1 : Espressioni regolari .....	26
4.3.2 : Trigger e procedure .....	26
<b>5   Manuale d'uso .....</b>	<b>30</b>
5.1 : Manuale Database .....	30
5.2 : Manuale GUI .....	30

## Capitolo 1

# Descrizione del Progetto

### 1.1 Analisi del problema

Si vuole mettere a punto, sfruttando come modello dei dati a supporto quello “Relazionale” una base di dati che possa contenere tutte le informazioni utili e necessarie al fine di poter gestire uno o più cantieri. Ma prima di ciò è necessario effettuare un analisi ad un livello più astratto, la quale a prescindere dalla tecnologia, ci permetta di modellare le entità in gioco per risolvere il problema.

In particolare, come è possibile evincere dalle specifiche, si avranno innanzitutto 4 entità le quali rappresentano : una la classe di oggetti che vogliamo gestire : i cantieri. E le restanti, 3 tipologie di dipendenti.

#### • *Amministratore*

Si occupa di svolgere l’azione principale del nostro sistema, cioè : l’apertura di un nuovo cantiere, nonché l’assegnazione di quest’ultimo in gestione ad un Capo-Cantiere. In più sarà dotato dei privilegi necessari a livello applicativo per poter inserire dati inerenti a dei sensori (che ancora dobbiamo definire) e modificarne la soglia di allarme.

#### • *CapoCantiere*

Incaricato da un amministratore, può gestire ed essere responsabile di uno o più cantieri. Definisce un ruolo per ogni operaio all’interno del cantiere di cui è responsabile e tiene traccia delle diverse aree del cantiere, alle quali assegna un operaio come responsabile. In più sarà dotato dei privilegi necessari a livello applicativo per poter leggere, filtrare, inserire i dati dei sensori, e modificarne la soglia di allarme.

### • *Operaio*

Ognuno ricopre un ruolo per ogni cantiere a cui è assegnato, tra i quali ci sarà un ruolo in particolare, quello di operatore, il quale avrà il compito di montare i sensori per le diverse aree del cantiere. Più in generale invece, ogni operaio può essere il responsabile di una o più aree di un cantiere, a seconda di quante gliene assegna il capo-cantiere.

Ognuno di questi tre ruoli identifica nella realtà dei fatti un dipendente dell'azienda, che sarà caratterizzato dai dati anagrafici ed alcuni recapiti email e telefonici.

Poi come analizzato prima ci sarà un entità centrale al nostro problema che sarà il cantiere :

### • *Cantiere*

Ogni cantiere sarà aperto, creato, da un amministratore, ed assegnato in gestione ad un capocantiere. Ogni cantiere può essere diviso in uno o più zone dette “Aree” che avranno diversi scopi e finalità a seconda del progetto del cantiere. Ed ogni area di un cantiere avrà una coppia di sensori.

Sensori di cui abbiamo già parlato ma non ancora definito, per cui ...

### • *Sensore*

Ogni sensore è montato da un operaio con ruolo operatore all'interno di un area di un cantiere, ed è responsabile della rilevazione di fughe di gas o rumori eccessivi. Saranno dotati di una soglia modificabile dagli amministratori e capo-cantieri, i quali potranno anche inserirne leggerne e filtrarne i dati all'interno del database.

Passiamo ora quindi alla cosiddetta fase di progettazione concettuale.

# **Capitolo 2**

# **Progettazione Concettuale**

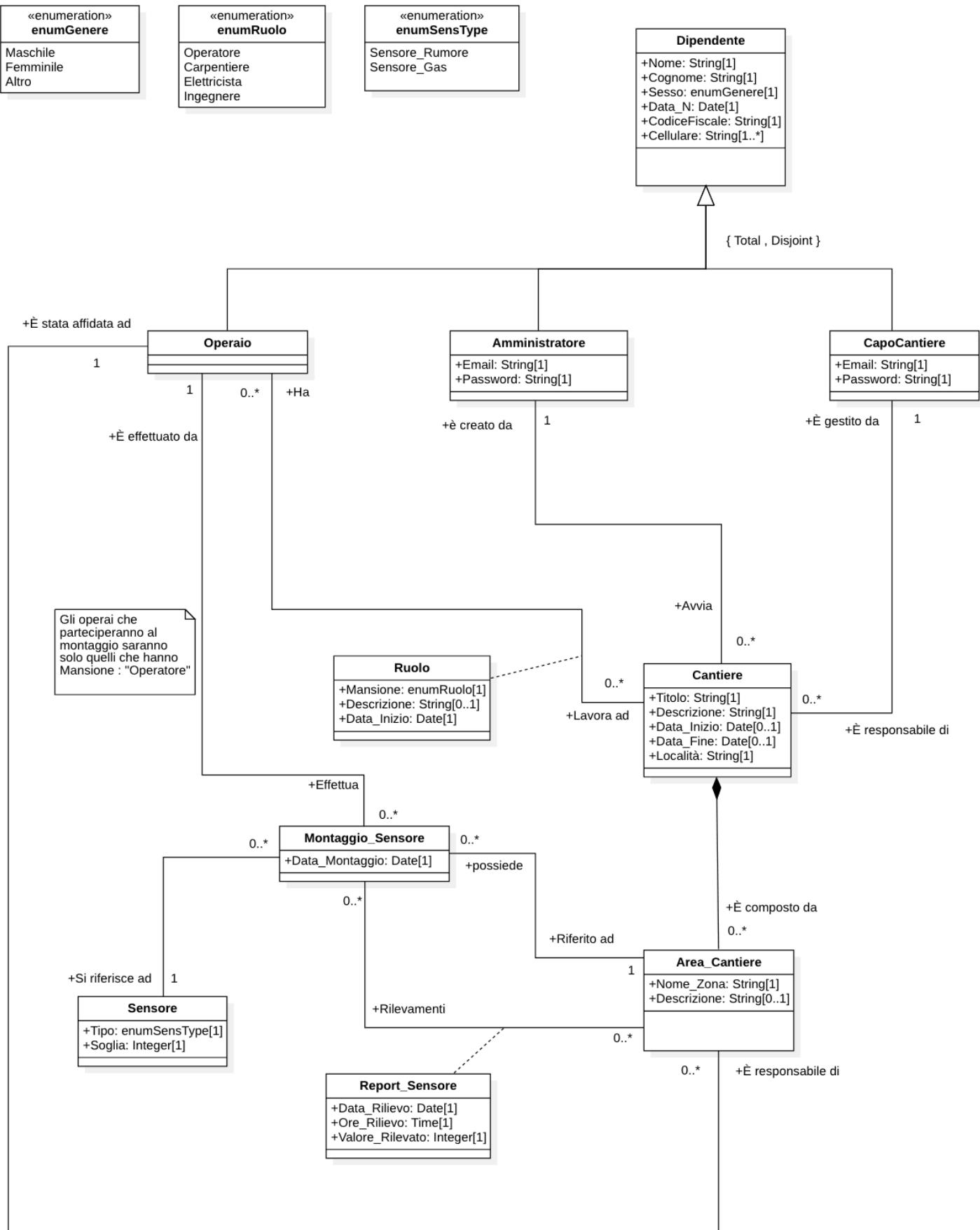
## ***2.1 Introduzione***

In questo capitolo inizia la progettazione della base di dati al livello di astrazione più alto. Dal risultato dell’analisi dei requisiti che devono essere soddisfatti si arriverà ad uno schema concettuale indipendente dalla struttura dei dati e dall’implementazione fisica: in tale schema concettuale, che verrà rappresentato usando un Class Diagram UML, si evidenzieranno le entità (concetti) rilevanti ai fini della rappresentazione dei dati e le relazioni che intercorrono tra esse; si delineeranno anche eventuali vincoli da imporre.

## ***2.2 Class Diagram***

A seguire il class diagram concettuale.

## Immagine 2.2.1 : Class diagram di analisi



## **2.3 Ristrutturazione 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. Al termine del procedimento il class diagram non conterrà attributi multipli, attributi strutturati, gerarchie di specializzazione ed eventualmente saranno eliminate anche associazioni uno ad uno tra classi, e attributi calcolabili.

### **2.3.1 Analisi gerarchie di specializzazioni**

All'interno del class diagram è presente un'unica gerarchia che è quella che riguarda la classe Dipendente con le classi Operaio, CapoCantiere ed Amministratore. Procederemo, dal momento che la specializzazione è di tipo totale disgiunto, includendo la superclasse 'Dipendente' all'interno delle diverse sottoclassi In modo da ottenere due cose:

- un diagramma molto più leggibile senza associazioni ricorsive o più associazioni tra le stesse classi.
- una netta distinzione tra i ruoli delle diverse istanze di dipendente senza dover fare un controllo su di un attributo discriminante.

### **2.3.2 Analisi delle ridondanze e attributi calcolabili**

Non sono state riscontrate ridondanze o attributi eventualmente calcolabili attraverso interrogazioni all'interno dello schema concettuale.

### **2.3.3 Analisi e rimozione di attributi multipli**

È presente all'interno del class diagram un attributo multiplo : il numero di cellulare dei dipendenti. Essendo una mole di informazioni relativamente piccola, e di scarsa importanza non andremo a creare un ulteriore entità per rappresentare uno o più numeri di telefono. Pertanto andremo semplicemente ad introdurre un ulteriore attributo opzionale per indicare eventualmente un secondo numero di telefono.

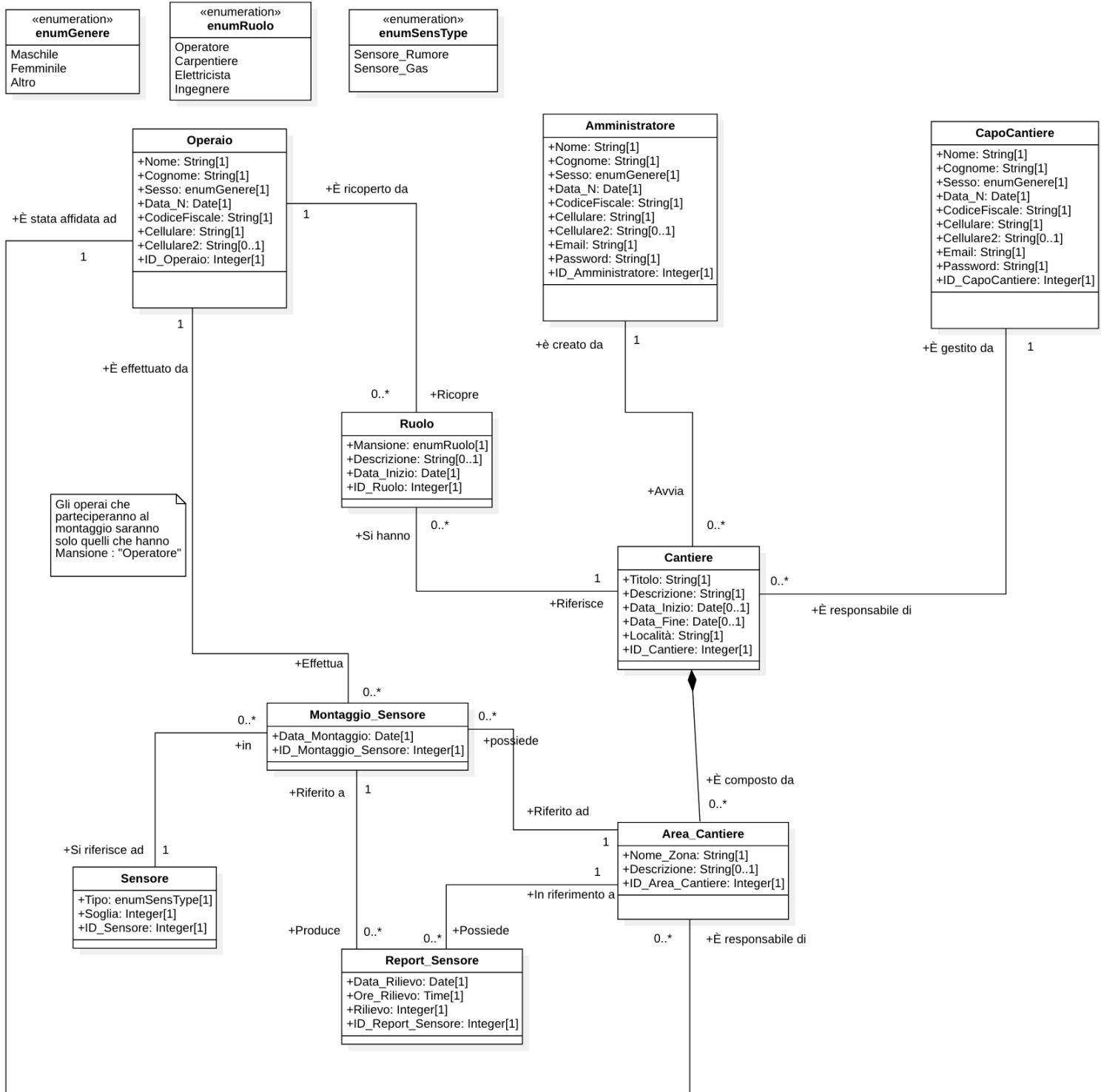
### **2.3.4 Analisi degli identificativi**

Da una prima analisi si potrebbe ipotizzare l'introduzione di chiavi surrogate generate automaticamente per ogni classe tranne che per i dipendenti che potrebbero utilizzare il codice fiscale, ma per uniformità e facilità di lettura/scrrittura delle interrogazioni introdurremo delle chiavi per tutte le classi i cui nomi rispetteranno il pattern : “ID\_<<NomeClasse>>”.

### **2.3.5 Analisi delle associazioni molti a molti e delle classi d'associazione**

Il modello relazionale non supporta la rappresentazioni di associazioni molti a molti, come ad esempio quella tra Operaio e Cantiere oppure tra Montaggio\_Sensore e Area\_Cantiere. Per di più, a tali associazioni abbiamo anche dovuto aggiungere una classe d'associazione per poter introdurre ulteriori attributi necessari. Pertanto all'interno della ristrutturazione andremo a sostituire queste due associazioni con due entità se stanti, e ognuna di esse avrà due associazioni uno a molti, che andranno a collegare le due classi coinvolte precedentemente alla nuova classe, che erediterà gli attributi della vecchia classe d'associazione.

## Immagine 2.3.1 : Class diagram Ristrutturato



## 2.4 Dizionari

Andremo ora ad introdurre delle tabelle utili alla comprensione del significato delle diverse classi, degli attributi che le caratterizzano e delle associazioni che intercorrono tra esse.

### 2.4.1 Dizionario delle Classi

Classe	Descrizione	Attributi
Cantiere	Descrittore di ciascun cantiere che la società prenderà in carico.	<b>ID_Cantiere (Integer)</b> : Chiave sintetica che permetterà di identificare in maniera univoca le singole istanze di cantieri. <b>Titolo (String)</b> : Titolo che sarà assegnato dall'amministratore al cantiere. <b>Descrizione (String)</b> : Breve descrizione opzionale riguardo il cantiere <b>Data_Inizio (Date)</b> : Data prevista di inizio dei lavori. <b>Data_Fine (Date)</b> : Data prevista fine lavori <b>Località (String)</b> : Comune e via dove verranno svolti i lavori.
Area_Cantiere	Descrittore delle diverse aree in cui può essere suddiviso un cantiere.	<b>ID_Area_Cantiere (Integer)</b> : Chiave sintetica che permetterà di identificare in maniera univoca le diverse aree dei diversi cantieri. <b>Nome_Zona (String)</b> : Nome da associare ad una singola zona (eg. zona residenziale). <b>Descrizione (String)</b> : Descrizione dell'area.

Classe	Descrizione	Attributi
<b>Amministratore</b>	Descrittore dei diversi amministratori in carica all'interno dell'azienda.	<p><b>ID_Amministratore (Integer)</b> : Chiave sintetica che permetterà di identificare univocamente un amministratore del sistema.</p> <p><b>Nome (String)</b> : Nome associato all'amministratore.</p> <p><b>Cognome (String)</b> : Cognome associato all'amministratore.</p> <p><b>Sesso (enumGenere)</b> : (eg. Uomo, Donna.)</p> <p><b>Data_N (Date)</b> : Data di nascita.</p> <p><b>CodiceFiscale (String)</b> : Codice Fiscale associato.</p> <p><b>Cellulare (String)</b> : Numero di cellulare amministratore.</p> <p><b>Cellulare2 (String)</b> : Secondo nonché facoltativo numero di cellulare associato.</p> <p><b>Email (String)</b> : Indirizzo email associato ad un amministratore per eventuali comunicazioni via e-mail.</p> <p><b>Password (String)</b> : Insieme all'id formate le credenziali di accesso al sistema associate all'amministratore.</p>
<b>CapoCantiere</b>	Descrittore dei diversi capo-cantieri responsabili dei cantieri.	<p><b>ID_CapoCantiere (Integer)</b> : Chiave sintetica che permetterà di identificare univocamente un capo cantiere all'intero del sistema.</p> <p><b>Nome (String)</b> : Nome associato al capo cantiere.</p> <p><b>Cognome (String)</b> : Cognome associato al capo cantiere.</p> <p><b>Sesso (enumGenere)</b> : (eg. Uomo, Donna.)</p> <p><b>Data_N (Date)</b> : Data di nascita.</p> <p><b>CodiceFiscale (String)</b> : Codice Fiscale associato.</p> <p><b>Cellulare (String)</b> : Numero di cellulare.</p> <p><b>Cellulare2 (String)</b> : Secondo nonché facoltativo numero di cellulare associato.</p> <p><b>Email (String)</b> : Indirizzo email associato ad un capo cantiere per eventuali comunicazioni via e-mail.</p> <p><b>Password (String)</b> : Insieme all'id formate le credenziali di accesso al sistema associate al capo cantiere.</p>

Classe	Descrizione	Attributi
Operario	Descrittore dei diversi operai a lavoro all'interno dell'azienda.	<p><b>ID_Operario (Integer)</b> : Chiave sintetica che permetterà di identificare univocamente un operario.</p> <p><b>Nome (String)</b> : Nome associato all'operario.</p> <p><b>Cognome (String)</b> : Cognome associato all'operario.</p> <p><b>Sesso (enumGenere)</b> : (eg. Uomo, Donna.)</p> <p><b>Data_N (Date)</b> : Data di nascita.</p> <p><b>CodiceFiscale (String)</b> : Codice Fiscale associato.</p> <p><b>Cellulare (String)</b> : Numero di cellulare di un operario.</p> <p><b>Cellulare2 (String)</b> : Secondo nonché facoltativo numero di cellulare associato.</p>
Ruolo	Descrittore dei diversi ruoli che può avere ogni operario in ogni cantiere.	<p><b>ID_Ruolo (Integer)</b> : Chiave sintetica che permetterà di identificare un singolo ruolo ricoperto da un operario all'interno di un singolo cantiere.</p> <p><b>Mansione (enumRuolo)</b> : Stringa che indica la mansione svolta da un operario all'interno di un cantiere. (eg. Operatore, Carpentiere, ...)</p> <p><b>Descrizione (String)</b> : Descrizione facoltativa della mansione da svolgere.</p> <p><b>Data_Inizio (Date)</b> : Data di inizio dello svolgimento della mansione.</p>
Sensore	Descrittore dei sensori di proprietà dell'azienda.	<p><b>ID_Sensore (Integer)</b> : Chiave sintetica associata ad ogni sensore utile ad identificarli univocamente.</p> <p><b>Tipo (enumSensType)</b> : Stringa che indica la tipologia di sensore. (eg. Sensore_Gas per rilevare fughe di gas)</p> <p><b>Soglia (Integer)</b> : Valore limite rilevabile dal sensore oltre il quale lanciare l'allarme.</p>
Montaggio_Sensore	Descrittore degli interventi di montaggio da parte degli operai di un sensore in un area cantiere.	<p><b>ID_Montaggio_Sensore (Integer)</b> : Chiave sintetica per identificare un singolo montaggio di un singolo sensore in un area di un cantiere da parte di un operatore.</p> <p><b>Data_Montaggio (Date)</b> : Data dell'avvenuto montaggio.</p>
Report_Sensore	Descrittore delle misurazioni effettuate dai sensori montati in una determinata area del cantiere	<p><b>ID_Report_Sensore (Integer)</b> : Chiave sintetica per identificare ogni report, ogni misurazione quindi effettuata dai sensori nelle aree cantiere.</p> <p><b>Data_Rilievo (Date)</b> : Data del rilievo da parte del sensore.</p> <p><b>Ore_Rilievo (Time)</b> : Ora del rilievo da parte del sensore.</p> <p><b>Rilievo (Integer)</b> : Valore rilevato dal sensore.</p>

## 2.4.2 Dizionario delle Associazioni

Associazione	Descrizione	Classi Coinvolte
<b>Amministratore   Cantiere</b>	Lega la creazione di un nuovo cantiere ad uno specifico amministratore.	<b>Amministratore [0..*]</b> , ruolo “ <b>avvia</b> ” : indica l’amministratore che ha creato un nuovo cantiere.  <b>Cantiere [1]</b> , ruolo “ <b>È creato da</b> ” : indica il nuovo cantiere immesso nel sistema da un amministratore.
<b>CapoCantiere   Cantiere</b>	Definisce le responsabilità di un cantiere legando la sua gestione ad un CapoCantiere.	<b>CapoCantiere [0..*]</b> , ruolo “ <b>È responsabile di</b> ” : indica il CapoCantiere delegato alla responsabilità un Cantiere.  <b>Cantiere [1]</b> , ruolo “ <b>È gestito da</b> ” : indica il cantiere in gestione da un CapoCantiere delegato.
<b>Cantiere   Area_Cantiere</b>	Esprime la suddivisione di una cantiere in diverse aree.	<b>Cantiere [0..*]</b> , ruolo “ <b>È composto da</b> ” : indica il Cantiere a cui si riferisce un’istanza di Area_Cantiere.  <b>Area_Cantiere [1]</b> : indica il cantiere a cui appartiene un’area del cantiere.
<b>Operaio   Area_Cantiere</b>	Definisce le responsabilità di un area di un cantiere legando la sua gestione ad un Operaio.	<b>Operaio [0..*]</b> , ruolo “ <b>È responsabile di</b> ” : indica l’Operaio delegato alla responsabilità di una singola o più aree di un cantiere.  <b>Area_Cantiere [1]</b> : indica l’area di un cantiere sotto la responsabilità di un operaio.
<b>Operaio   Ruolo</b>	Lega un operaio ad un determinato ruolo.	<b>Operaio [0..*]</b> , ruolo “ <b>Ricopre</b> ” : indica l’Operaio che ricopre un determinato ruolo.  <b>Ruolo [1]</b> , ruolo “ <b>È ricoperto da</b> ” : indica il ruolo a cui è assegnato un singolo operaio.
<b>Ruolo   Cantiere</b>	Definisce a quale cantiere si riferisce un ruolo ricoperto da un operaio.	<b>Ruolo [1]</b> , ruolo “ <b>Riferisce</b> ” : indica un ruolo a quale cantiere appartiene e si riferisce.  <b>Cantiere [0..*]</b> , ruolo “ <b>Si hanno</b> ” : indica un cantiere in cui si avranno più ruoli.
<b>Sensore   Montaggio_Sensore</b>	Indica quale sensore è stato montato legando un sensore ad una istanza di montaggio.	<b>Sensore [0..*]</b> , ruolo “ <b>in</b> ” : indica il sensore di un montaggio.  <b>Montaggio_Sensore [1]</b> , ruolo “ <b>Si riferisce ad</b> ” : indica a quale montaggio si riferisce un sensore.

Associazione	Descrizione	Classi Coinvolte
<b>Operaio</b> — <b>Montaggio_Sensore</b>	Indica chi è stato ad eseguire il montaggio di un sensore.	<b>Operaio [0..*]</b> , ruolo “ <b>Effettua</b> ” : indica l’Operaio che ha partecipato ad un montaggio.
<b>Area_Cantiere</b> — <b>Montaggio_Sensore</b>	Indica in quale area di qualche cantiere è stato effettuato il montaggio.	<b>Montaggio_Sensore [1]</b> , ruolo “ <b>È effettuato da</b> ” : Indica i montaggi effettuati da un Operaio.
<b>Montaggio_Sensore</b> — <b>Report_Sensore</b>	Lega un report riguardo i rilievi effettuati ad un sensore montato.	<b>Area_Cantiere [0..*]</b> , ruolo “ <b>Possiede</b> ” : indica l’area di un cantiere a cui si riferisce un montaggio effettuato da un operaio.
<b>Report_Sensore</b> — <b>Area_Cantiere</b>	Lega un report riguardo i rilievi effettuati ad un area di qualche cantiere.	<b>Montaggio_Sensore [1]</b> , ruolo “ <b>Riferito ad</b> ” : indica un montaggio sensore a quale area cantiere si riferisce.
<b>Report_Sensore</b> — <b>Area_Cantiere</b>	Lega un report riguardo i rilievi effettuati ad un area di qualche cantiere.	<b>Montaggio_Sensore [0..*]</b> , ruolo “ <b>Produce</b> ” : indica da quale sensore montato deriva un report.
<b>Report_Sensore</b> — <b>Area_Cantiere</b>	Lega un report riguardo i rilievi effettuati ad un area di qualche cantiere.	<b>Report_Sensore [1]</b> , ruolo “ <b>Riferito a</b> ” : indica i report riferiti ad un sensore.
<b>Report_Sensore</b> — <b>Area_Cantiere</b>	Lega un report riguardo i rilievi effettuati ad un area di qualche cantiere.	<b>Report_Sensore [1]</b> , ruolo “ <b>In riferimento a</b> ” : indica i report in riferimento ad un area cantiere.
<b>Report_Sensore</b> — <b>Area_Cantiere</b>	Lega un report riguardo i rilievi effettuati ad un area di qualche cantiere.	<b>Area_Cantiere [0..*]</b> , ruolo “ <b>Possiede</b> ” : indica il possesso per ogni area cantiere di uno o più report di rilievo da parte di sensori montati.

## 2.4.3 Dizionario dei vincoli

Nome del vincolo	Tipo	Descrizione
<b>Check_CF</b>	<i>Dominio</i>	Il codice fiscale laddove presente in una classe (Amministratore, CapoCantiere oppure Operaio) dovrà rispettare l'espressione regolare formata da 16 caratteri che prevede : <ul style="list-style-type: none"> <li>• 6 caratteri alfabetici (Cognome e Nome)</li> <li>• 2 caratteri numerici (Anno di Nascita)</li> <li>• 1 carattere alfabetico (Mese di Nascita)</li> <li>• 2 caratteri numerici (Giorno e sesso)</li> <li>• 4 caratteri, 1 alfabetico e 3 numerici (Codice nazionale comune/stato estero)</li> <li>• 1 carattere alfabetico di controllo</li> </ul>
<b>Check_Email</b>	<i>Dominio</i>	L'Email utilizzata per effettuare l'accesso dagli amministratori e capo cantieri dovrà rispettare il formato standard per un email eg. (esempio@email.com)
<b>UniqueCF</b>	<i>Intrarelazionale</i>	Il codice fiscale laddove presente in una tabella (Amministratore, CapoCantiere oppure Operaio) dovrà essere unico. Sarà possibile invece inserire due profili diversi in due tabelle diverse (ad esempio un amministratore ed un capo cantiere) con gli stessi codici fiscali
<b>CheckEmailUnica</b>	<i>Interrelazionale</i>	L'Email utilizzata per effettuare l'accesso dagli amministratori e capo cantieri dovrà essere unica all'interno di tutto il database. Non sarà mai possibile inserire un amministratore ed un capo cantiere con la stessa email.
<b>NomeUnico</b>	<i>Intrarelazionale</i>	Il nome utilizzato per le aree cantiere dovrà essere univoco.
<b>CkMansione</b>	<i>Interrelazionale</i>	Soltanto gli operai con mansione da "Operatore" all'interno di un cantiere potranno montare i sensori nelle diverse aree/zona.
<b>CkRuoloOperaio</b>	<i>Interrelazionale</i>	Un area cantiere si può assegnare in gestione soltanto ad operai con un ruolo all'interno del cantiere a cui appartiene l'area.
<b>EnumRuolo</b>	<i>Dominio</i>	È possibile scegliere come mansione per un ruolo tra 4 possibili posizioni lavorative : Ingegnere, Elettricista, Carpentiere, Operatore
<b>UnicoRuolo</b>	<i>Intrarelazionale</i>	Non è possibile assegnare allo stesso operaio, due ruoli, nello stesso cantiere.
<b>EnumSensType</b>	<i>Dominio</i>	I sensori potranno essere scelti tra due categorie : quelli che rilevano fughe di gas, e quelli che rilevano rumori.
<b>CheckNumSens</b>	<i>Interrelazionale</i>	Potranno essere montati al più 2 sensori per area cantiere.
<b>UnicoTitolo</b>	<i>Intrarelazionale</i>	Il titolo assegnato ad un cantiere dovrà essere univoco.

Nome del vincolo	Tipo	Descrizione
<b>CKUnicoCantiereGestito</b>	Interrelazionale	Sarà possibile affidare ai capo cantieri un unico cantiere in gestione per volta.(I lavori all'interno di un cantiere verranno considerati terminati al superamento della data di fine lavori)

# Capitolo 3

# Progettazione Logica

## 3.1 Introduzione

In questo capitolo sarà trattata la fase successiva della progettazione del database scendendo ad un livello di astrazione intermedio in cui terremo conto del modello dei dati adottato, in questo caso il modello relazionale, per andare a tradurre lo schema concettuale ristrutturato, già predisposto alla traduzione, in uno schema logico. Come di consueto all'interno di questi ultimi le chiavi primarie sono indicate con una singola sottolineatura mentre le chiavi esterne con una doppia sottolineatura.

## 3.2 Schema Logico

---

**Amministratore** ( Nome, Cognome, Sesso, Data\_N, CodiceFiscale, Cellulare, Cellulare2, Email, Password, ID\_Amministratore )

---

**CapoCantiere** ( Nome, Cognome, Sesso, Data\_N, CodiceFiscale, Cellulare, Cellulare2, Email, Password, ID\_CapoCantiere )

---

**Cantiere** ( Titolo, Descrizione, Data\_Inizio, Data\_Fine, Località, ID\_Cantiere, ID\_Creatore, ID\_Responsabile )

---

Chiavi esterne      ID\_Creatore      —> Amministratore.ID\_Amministratore  
                        ID\_Responsabile      —> CapoCantiere.ID\_CapoCantiere

---

**Operaio** ( Nome, Cognome, Sesso, Data\_N, CodiceFiscale, Cellulare, Cellulare2, ID\_Operaio )

---

**Ruolo** ( Mansione, Descrizione, Data\_Inizio, ID\_Ruolo, ID\_Lavoratore, ID\_CantiereLavori )

---

Chiavi esterne      ID\_Lavoratore      —> Operaio.ID\_Operaio  
                        ID\_CantiereLavori      —> Cantiere.ID\_Cantiere

**Area\_Cantiere** ( Nome\_Zona, Descrizione, ID\_Area\_Cantiere, ID\_Gestore,  
ID\_CantiereAppartenenza )

Chiavi esterne      ID\_Gestore      —>      Operaio.ID\_Operaio  
                        ID\_CantiereAppartenenza      —>      Cantiere.ID\_Cantiere

---

**Sensore** ( Tipo, Soglia, ID\_Sensore )

---

**Montaggio\_Sensore** ( Data\_Montaggio, ID\_Montaggio, ID\_Sensore, ID\_Area, ID\_Operatore, )

Chiavi esterne      ID\_Sensore      —>      Sensore.ID\_Sensore  
                        ID\_Area      —>      Area\_Cantiere.ID\_Area\_Cantiere  
                        ID\_Operatore      —>      Operaio.ID\_Operaio

---

**Report\_Sensore** ( Data\_Rilievo, Ore\_Rilievo, Rilievo, ID\_Report\_Sensore, ID\_SensMontato,  
ID\_Area\_Rilievo )

Chiavi esterne      ID\_SensMontato      —>      Montaggio\_Sensore.ID\_Montaggio  
                        ID\_AreaRilievo      —>      Area\_Cantiere.ID\_Area\_Cantiere

---

# Capitolo 4

# Progettazione Fisica

## 4.1 Introduzione

In questo capitolo ci avviamo verso la fase di progettazione fisica, quella fase implementativa dove prendiamo in considerazione un DBMS in particolare, PostgreSQL, con la sua sintassi, per fornire il codice che andrà a definire le tabelle secondo lo schema logico fornito precedentemente, le associazioni, i vincoli ed eventuali procedure o trigger da eseguire. Iniziamo creando il nostro database sul dbms da noi scelto.

```
1 CREATE DATABASE GestioneCantieri_UniNa ;
```

## 4.2 Definizione delle tabelle

A seguire gli estratti SQL per la creazione delle tabelle definite nello schema logico.

### 4.2.1 Definizione della tabella ‘Amministratore’

```
1 CREATE TABLE Amministratore (
2     Nome VARCHAR(50) NOT NULL,
3     Cognome VARCHAR(50) NOT NULL,
4     Sesso VARCHAR(1) NOT NULL,
5     Data_n DATE NOT NULL,
6     CodiceFiscale VARCHAR(16) NOT NULL,
7     Cellulare VARCHAR(15) NOT NULL,
8     Cellulare2 VARCHAR(15),
9     Email VARCHAR(50) NOT NULL,
10    pswrd VARCHAR(100) NOT NULL,
11    ID_Amministratore INTEGER,
12
13    CONSTRAINT Amministratore_PK PRIMARY KEY(ID_Amministratore),
14    CONSTRAINT enumGenere CHECK (Sesso IN ('M', 'F')),
15    CONSTRAINT email_amm UNIQUE (Email),
16    CONSTRAINT cf_amm UNIQUE (CodiceFiscale)
17 );
```

#### **4.2.2 Definizione della tabella ‘CapoCantiere’**

```
1 CREATE TABLE CapoCantiere (
2     Nome VARCHAR(50) NOT NULL,
3     Cognome VARCHAR(50) NOT NULL,
4     Sesso VARCHAR(1) NOT NULL,
5     Data_n DATE NOT NULL,
6     CodiceFiscale VARCHAR(16) NOT NULL,
7     Cellulare VARCHAR(15) NOT NULL,
8     Cellulare2 VARCHAR(15),
9     Email VARCHAR(50) NOT NULL,
10    pswrd VARCHAR(100) NOT NULL,
11    ID_CapoCantiere INTEGER,
12
13    CONSTRAINT CapoCantiere_PK PRIMARY KEY(ID_CapoCantiere),
14    CONSTRAINT enumGenere CHECK (Sesso IN ('M','F')),
15    CONSTRAINT email_cc UNIQUE (Email),
16    CONSTRAINT cf_cc UNIQUE (CodiceFiscale)
17 );
```

#### **4.2.3 Definizione della tabella ‘Operaio’**

```
1 CREATE TABLE Operaio (
2     Nome VARCHAR(50) NOT NULL,
3     Cognome VARCHAR(50) NOT NULL,
4     Sesso VARCHAR(1) NOT NULL,
5     Data_n DATE NOT NULL,
6     CodiceFiscale VARCHAR(16) NOT NULL,
7     Cellulare VARCHAR(15) NOT NULL,
8     Cellulare2 VARCHAR(15),
9     ID_Operaio INTEGER,
10
11    CONSTRAINT Operaio_PK PRIMARY KEY(ID_Operaio),
12    CONSTRAINT enumGenere CHECK (Sesso IN ('M','F')),
13    CONSTRAINT cf_op UNIQUE (CodiceFiscale)
14 );
```

#### **4.2.4 Definizione della tabella ‘Cantiere’**

```
1 CREATE TABLE Cantiere (
2     Titolo VARCHAR(100) NOT NULL,
3     Descrizione VARCHAR(100) NOT NULL,
4     Data_Inizio DATE,
5     Data_Fine DATE,
6     Localita VARCHAR(100) NOT NULL,
7     ID_Cantiere INTEGER,
8     ID_Creatore INTEGER NOT NULL,
9     ID_Responsabile INTEGER NOT NULL,
10
11    CONSTRAINT Cantiere_PK PRIMARY KEY(ID_Cantiere),
12
13    CONSTRAINT Cantiere_FK1 FOREIGN KEY (ID_Creatore)
14    REFERENCES Amministratore(ID_Amministratore) ON DELETE NO ACTION,
15
16    CONSTRAINT Cantiere_FK2 FOREIGN KEY (ID_Responsabile)
17    REFERENCES CapoCantiere(ID_CapoCantiere) ON DELETE NO ACTION,
18
19    CONSTRAINT Utitolo UNIQUE (Titolo)
20 );
```

#### **4.2.5 Definizione della tabella ‘Sensore’**

```
1 CREATE TABLE Sensore (
2     ID_Sensore INTEGER,
3     Tipo VARCHAR(20) NOT NULL,
4     Soglia INTEGER NOT NULL,
5
6     CONSTRAINT Sensore_PK PRIMARY KEY(ID_Sensore),
7
8     CONSTRAINT enumSensType CHECK (Tipo IN ('Sensore_Rumore', 'Sensore_Gas'))
9 );
```

#### **4.2.6 Definizione della tabella ‘Ruolo’**

```
1 CREATE TABLE Ruolo (
2     Mansione VARCHAR(30) NOT NULL,
3     Descrizione VARCHAR(50),
4     Data_Inizio DATE NOT NULL,
5     ID_Ruolo INTEGER,
6     ID_Lavoratore INTEGER NOT NULL,
7     ID_CantiereLavori INTEGER NOT NULL,
8
9     CONSTRAINT Ruolo_PK PRIMARY KEY(ID_Ruolo),
10
11    CONSTRAINT Ruolo_FK1 FOREIGN KEY (ID_Lavoratore)
12        REFERENCES Operaio(ID_Operaio) ON DELETE NO ACTION ON UPDATE CASCADE,
13
14    CONSTRAINT Ruolo_FK2 FOREIGN KEY (ID_CantiereLavori)
15        REFERENCES Cantiere(ID_Cantiere) ON DELETE NO ACTION ON UPDATE CASCADE,
16
17    CONSTRAINT enumRuolo CHECK (Mansione IN ('Operatore','Carpentiere','Elettricista','Ingegnere')),
18
19    CONSTRAINT unicoRuolo UNIQUE (ID_Lavoratore,ID_CantiereLavori)
20 );
```

#### **4.2.7 Definizione della tabella ‘Area\_Cantiere’**

```
1 CREATE TABLE Area_Cantiere (
2     Nome_Zona VARCHAR(50) NOT NULL,
3     Descrizione VARCHAR(50),
4     ID_Area_Cantiere INTEGER,
5     ID_Gestore INTEGER NOT NULL,
6     ID_CantiereAppartenenza INTEGER NOT NULL,
7
8     CONSTRAINT Area_Cantiere_PK PRIMARY KEY(ID_Area_Cantiere),
9
10    CONSTRAINT Area_Cantiere_FK1 FOREIGN KEY (ID_Gestore)
11        REFERENCES Operaio(ID_Operaio) ON DELETE NO ACTION ON UPDATE CASCADE,
12
13    CONSTRAINT Area_Cantiere_FK2 FOREIGN KEY (ID_CantiereAppartenenza)
14        REFERENCES Cantiere(ID_Cantiere) ON DELETE NO ACTION ON UPDATE CASCADE,
15
16    CONSTRAINT NomeUnico UNIQUE (Nome_Zona)
17 );
```

#### **4.2.8 Definizione della tabella ‘Montaggio\_Sensore’**

```
1 CREATE TABLE Montaggio_Sensore (
2     Data_Montaggio DATE NOT NULL,
3     ID_Montaggio INTEGER,
4     ID_Sensore INTEGER NOT NULL,
5     ID_Area INTEGER NOT NULL,
6     ID_Operatore INTEGER NOT NULL,
7
8     CONSTRAINT Montaggio_Sensore_PK PRIMARY KEY(ID_Montaggio),
9
10    CONSTRAINT Montaggio_Sensore_FK1 FOREIGN KEY (ID_Sensore)
11        REFERENCES Sensore(ID_Sensore) ON DELETE NO ACTION ON UPDATE CASCADE,
12
13    CONSTRAINT Montaggio_Sensore_FK2 FOREIGN KEY (ID_Area)
14        REFERENCES Area_Cantiere(ID_Area_Cantiere) ON DELETE NO ACTION ON UPDATE CASCADE,
15
16    CONSTRAINT Montaggio_Sensore_FK3 FOREIGN KEY (ID_Operatore)
17        REFERENCES Operaio(ID_Operaio) ON DELETE NO ACTION ON UPDATE CASCADE
18 );
```

#### **4.2.9 Definizione della tabella ‘Report\_Sensore’**

```
1 CREATE TABLE Report_Sensore (
2     Data_Rilievo DATE NOT NULL,
3     Ore_Rilievo TIME NOT NULL,
4     Rilievo INTEGER NOT NULL,
5     ID_Report_Sensore INTEGER,
6     ID_SensMontato INTEGER NOT NULL,
7     ID_Area_Rilievo INTEGER NOT NULL,
8
9     CONSTRAINT Report_Sensore_PK PRIMARY KEY(ID_Report_Sensore),
10
11    CONSTRAINT Report_Sensore_FK1 FOREIGN KEY (ID_SensMontato)
12        REFERENCES Sensore(ID_Sensore) ON DELETE NO ACTION ON UPDATE CASCADE,
13
14    CONSTRAINT Report_Sensore_FK2 FOREIGN KEY (ID_Area_Rilievo)
15        REFERENCES Area_Cantiere(ID_Area_Cantiere) ON DELETE NO ACTION ON UPDATE CASCADE
16 );
```

## 4.3 Definizione procedure trigger e regexp

A questo punto, definite le tabelle ed i relativi vincoli intra-relazionali quali vincoli di dominio, di unicità ed n-upla, non ci resta che definire i trigger che ci occorrono al fine di garantire i diversi vincoli inter-relazionali in precedenza specificati all'interno del dizionario, nonché i vincoli che controlleranno il rispettare o meno di determinate espressioni regolari.

### 4.3.1 Espressioni regolari

Andremo di seguito a definire due espressioni regolari che dovranno essere rispettate per quanto riguarda l'inserimento di email e codice fiscale.

```
1 ALTER TABLE amministratore
2 ADD CONSTRAINT check_Email CHECK (Email ~* '^@[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+[.][A-Za-z]+$');
3
4 ALTER TABLE capocantiere
5 ADD CONSTRAINT check_Email CHECK (Email ~* '^@[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+[.][A-Za-z]+$');
6
7 ALTER TABLE amministratore
8 ADD CONSTRAINT check_CF CHECK (codicefiscale ~* '^@[A-Z]{3}[A-Z]{3}[0-9]{2}[A-EHLMR-T][0-9]{2}[A-Z][0-9]{3}[A-Z]$');
9
10 ALTER TABLE capocantiere
11 ADD CONSTRAINT check_CF CHECK (codicefiscale ~* '^@[A-Z]{3}[A-Z]{3}[0-9]{2}[A-EHLMR-T][0-9]{2}[A-Z][0-9]{3}[A-Z]$');
12
13 ALTER TABLE operaio
14 ADD CONSTRAINT check_CF CHECK (codicefiscale ~* '^@[A-Z]{3}[A-Z]{3}[0-9]{2}[A-EHLMR-T][0-9]{2}[A-Z][0-9]{3}[A-Z]$');
```

### 4.3.2 Trigger e Procedure

A causa di vincoli imposti dal DBMS scelto, Postgres, i trigger non faranno altro che attivarsi al verificarsi di una determinata condizione e invece che eseguire direttamente codice plsql, richiamano a loro volta una procedura che conterrà il vero e proprio codice. Di seguito le loro definizioni :

```
1 CREATE TRIGGER TriggerControllaEmailAmm
2 BEFORE INSERT OR UPDATE OF email ON amministratore
3 FOR EACH ROW
4 EXECUTE PROCEDURE CKEmailUnica();
5
6 CREATE TRIGGER TriggerControllaEmailCC
7 BEFORE INSERT OR UPDATE OF email ON capocantiere
8 FOR EACH ROW
9 EXECUTE PROCEDURE CKEmailUnica();
```

```

1 CREATE FUNCTION CKEmailUnica()
2 RETURNS trigger
3 AS $CKEmailUnica$
4 BEGIN
5     IF NOT EXISTS ( SELECT *
6                     FROM amministratore
7                     WHERE amministratore.email LIKE NEW.email )
8     THEN
9         IF NOT EXISTS ( SELECT *
10                     FROM capocantiere
11                     WHERE capocantiere.email LIKE NEW.email )
12         THEN
13             RETURN NEW;
14         ELSE
15             RAISE NOTICE 'Email già esistente nel database';
16             RETURN NULL;
17         END IF;
18     ELSE
19         RAISE NOTICE 'Email già esistente nel database';
20         RETURN NULL;
21     END IF;
22 END
23 $CKEmailUnica$
24 LANGUAGE 'plpgsql';

```

```

1 CREATE TRIGGER TriggerNumeroSensoriMontati
2 BEFORE INSERT ON montaggio_sensore
3 FOR EACH ROW
4 EXECUTE PROCEDURE CKNumSens();
5
6 CREATE FUNCTION CKNumSens()
7 RETURNS trigger
8 AS $CKNumSens$
9 BEGIN
10    IF ( 2 > (SELECT COUNT(*) FROM montaggio_sensore JOIN area_cantiere
11                ON montaggio_sensore.id_area = area_cantiere.id_area_cantiere WHERE id_area_cantiere = NOW.id_area) )
12    THEN
13        RETURN NEW ;
14    ELSE
15        RAISE NOTICE 'Sono presenti già due sensori montati all'interno dell'area';
16        RETURN NULL;
17    END IF;
18 END
19 $CKNumSens$
20 LANGUAGE 'plpgsql';

```

```

1 CREATE TRIGGER TriggerSingoloCantiereInGestione
2 BEFORE INSERT OR UPDATE OF id_responsabile ON cantiere
3 FOR EACH ROW
4 EXECUTE PROCEDURE CKUnicoCantiereGestito();
5
6 CREATE FUNCTION CKUnicoCantiereGestito()
7 RETURNS trigger
8 AS $CKUnicoCantiereGestito$
9 BEGIN
10 IF NOT EXISTS ( SELECT *
11             FROM cantiere
12             WHERE id_responsabile = NEW.id_responsabile )
13 THEN
14     RETURN NEW ;
15 ELSE
16     RAISE NOTICE 'Si può assegnare in gestione un unico cantiere per volta';
17     RETURN NULL;
18 END IF;
19 END
20 $CKUnicoCantiereGestito$
21 LANGUAGE 'plpgsql';

```

```

1 CREATE TRIGGER TriggerControllaRuoloOperaio
2 BEFORE INSERT OR UPDATE OF id_gestore ON area_cantiere
3 FOR EACH ROW
4 EXECUTE PROCEDURE CKRuoloOperaio();
5
6 CREATE FUNCTION CKRuoloOperaio()
7 RETURNS trigger
8 AS $CKRuoloOperaio$
9 BEGIN
10 IF EXISTS ( SELECT *
11             FROM ruolo
12             WHERE id_lavoratore = NEW.id_gestore AND id_cantierelavori = NEW.id_cantiereappartenenza )
13 THEN
14     RETURN NEW ;
15 ELSE
16     RAISE NOTICE 'Si può assegnare in gestione un area cantiere soltanto ad operai con
17                 un ruolo all'interno del cantiere a cui appartiene la zona';
18     RETURN NULL;
19 END IF;
20 END
21 $CKRuoloOperaio$
22 LANGUAGE 'plpgsql';

```

```

1 CREATE TRIGGER TriggerControllaMansione
2 BEFORE INSERT ON montaggio_sensore
3 FOR EACH ROW
4 EXECUTE PROCEDURE CKMansione();
5
6 CREATE FUNCTION CKMansione()
7 RETURNS trigger
8 AS $CKMansione$
9 BEGIN
10    IF EXISTS ( SELECT *
11                  FROM ruolo
12                  WHERE (ruolo.id_lavoratore = NEW.id_operatore AND ruolo.mansione LIKE 'Operatore' AND
13                         ruolo.id_cantierelavori = (SELECT id_cantiereappartenenza
14                                         FROM area_cantiere
15                                         WHERE id_area_cantiere = NEW.id_area) ) )
16    THEN
17        RETURN NEW ;
18    ELSE
19        RAISE NOTICE 'Soltanto gli operai con una mansione da operatore nel cantiere possono montare i sensori';
20        RETURN NULL;
21    END IF;
22 END
23 $CKMansione$
24 LANGUAGE 'plpgsql';
--
```

# Capitolo 5

# Manuale d'uso

## 5.1 Manuale Database

Tra i file riguardanti il progetto ci sono due cartelle denominate rispettivamente DB\_BackupTar e DB\_BackupPlain le quali conterranno rispettivamente due backup dello stesso database in due dei possibili formati che è possibile esportare. Basterà ripristinare da uno dei due file per avere un'immagine completa della base di dati. All'interno del database saranno presenti dei dati di esempio al fine di simulare la messa in opera ed il funzionamento del software.

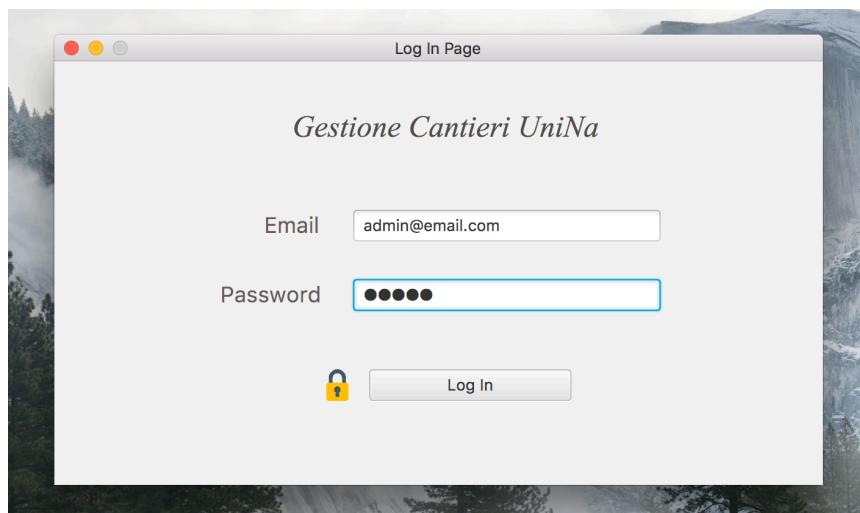
## 5.2 Manuale GUI

In questo paragrafo andremo attraverso diversi screenshot a fare una rapida panoramica di come funziona e come accedere per la prima volta al sistema.

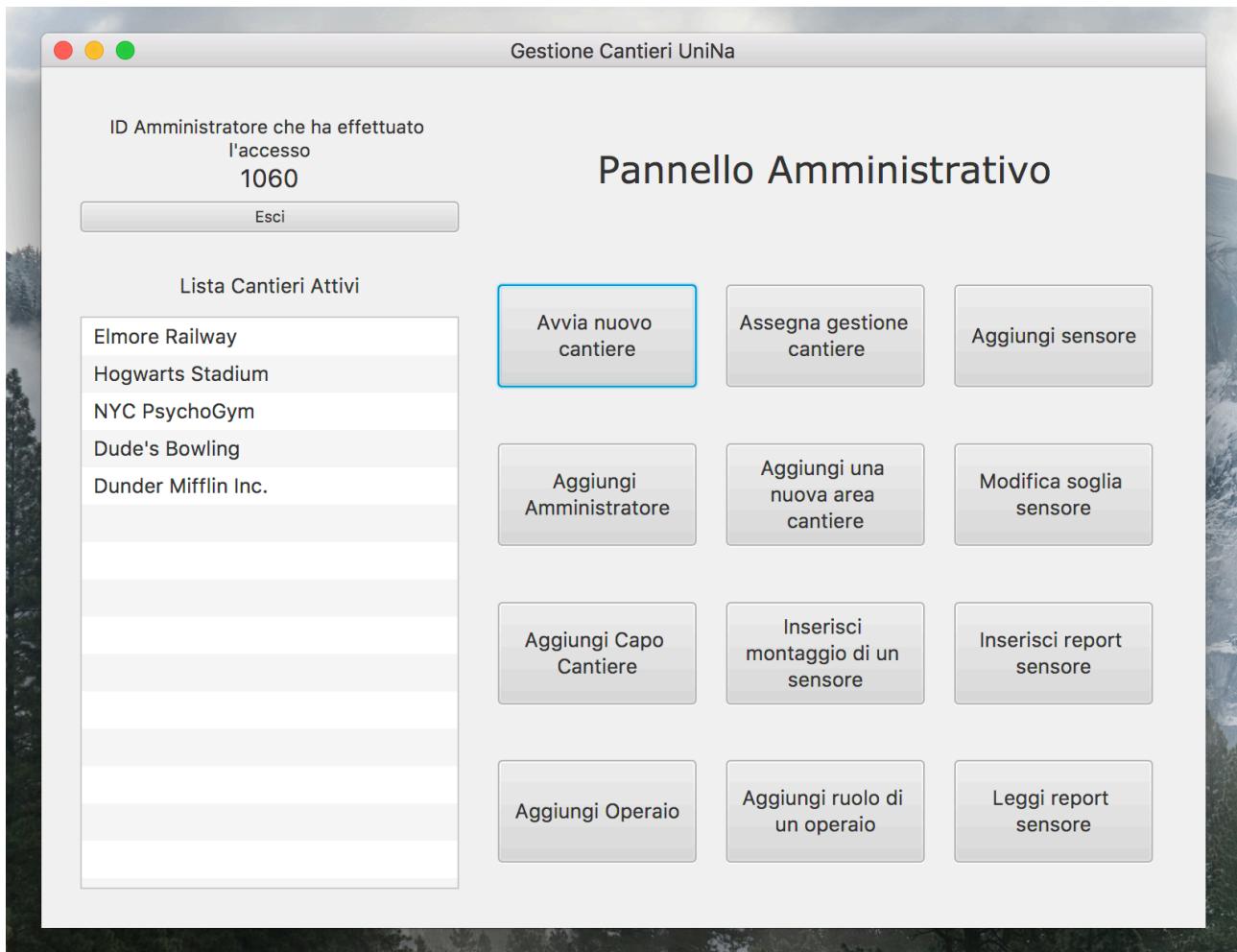
Innanzitutto è messo a disposizione del committente un account le cui credenziali permetteranno l'accesso nelle vesti di amministratore del sistema. Le credenziali in questione andranno inserite nella pagina di login che comparirà all'avvio del software, e sono :

Email : admin@email.com

Password : admin

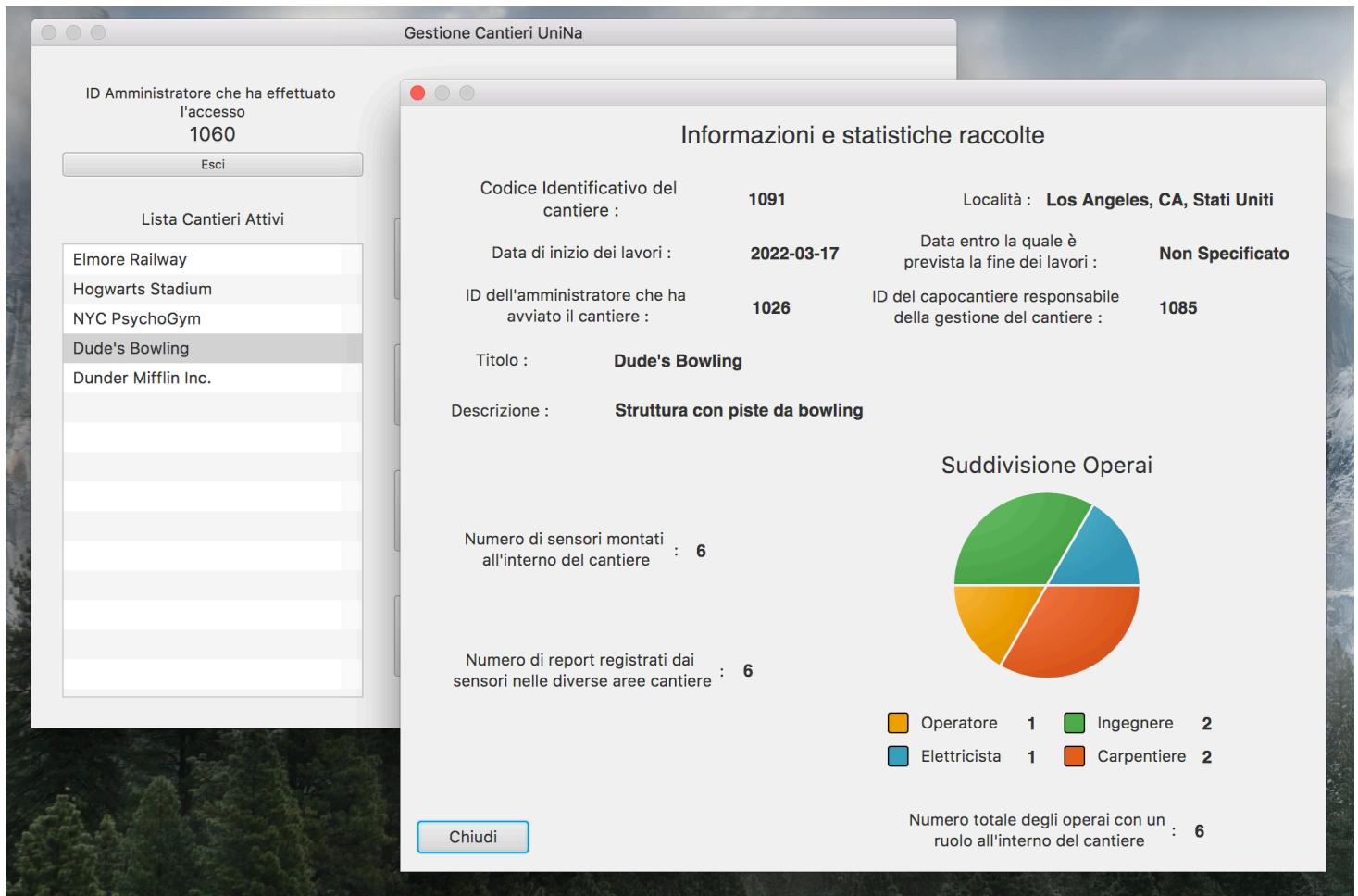


A questo punto non ci resta che cliccare su login e verremo indirizzati verso quest'altra finestra :



All'interno di questa area sarà possibile attraverso i pulsanti, ai quali è stata applicata un etichetta quanto più auto-esplicativa possibile, effettuare tutto ciò che è richiesto dalle specifiche del committente tra cui ad esempio aggiungere nuovi operai all'impresa, o aggiungere nuovi profili amministrativi/capocantieri, avviare nuovi cantieri, e così via.

Inoltre dal menu laterale sarà possibile, cliccando sul nome di un cantiere in corso, poter visualizzare varie statistiche riguardo quel cantiere come ad esempio il numero e la suddivisione degli operai al suo interno come illustrato di seguito.



Quanto detto finora vale anche per i profili dei capo cantieri i quali dovranno semplicemente inserire la loro email così come gli amministratori ed effettuare l'accesso tramite la stessa interfaccia. Sarà il software a riconoscere se si tratta di un accesso da parte di un amministratore o da parte di un capocantiere.

Le uniche differenze si trovano nel tipo di informazioni che potranno leggere i capo cantieri e gli amministratori dai loro rispettivi pannelli laterali. Gli amministratori vedranno informazioni raccolte sui cantieri, i capo cantieri invece monitoreranno gli operai che ricoprono un ruolo all'interno del cantiere da loro gestito.

Infine, non è stata prevista la possibilità di effettuare registrazioni di nuovi profili se non dall'interno del software stesso tramite un altro amministratore, come quello fornito di default.