Relazione

Progetto BDSI

Autori:

**Daniil Radchanka (7079901)**

**Andrea Rolleri (7037369)**

**Anastasia Moskalenko (7015595)**

Indice

[Richiesta 3](#_Toc139968925)

[Progettazione concettuale 4](#_Toc139968926)

[Glossario dei termini 4](#_Toc139968927)

[Costruzione dello schema concettuale 5](#_Toc139968928)

[Progettazione logica 9](#_Toc139968929)

[Eliminazione di attributi composti 9](#_Toc139968930)

[Eliminazione delle gerarchie 9](#_Toc139968931)

[Traduzione verso il modello relazionale 12](#_Toc139968932)

[Progettazione fisica 13](#_Toc139968933)

[Struttura del file assegnati al progetto 13](#_Toc139968934)

[Tabelle e viste 13](#_Toc139968935)

[Procedure 13](#_Toc139968936)

[Trigger 13](#_Toc139968937)

# Richiesta

Il Sistema Sanitario Internazionale (S.S.I) vuole implementare un sito per gestire le cliniche e chiede implementare una base di dati per questo. Lo scopo principale di questo sito è di mantenere i dati delle cliniche e proporre la prenotazione di visite da parte di pazienti.

Il sito deve prevedere la possibilità di accesso di dottori, amministratori e pazienti tramite i loro account. Per avere l’accesso al sito bisogna inserire il login e password.

I pazienti hanno codice fiscale, nome, cognome, età, indirizzo di residenza che è composto dalla via, città, nazione e cap. Loro possono effettuare la prenotazione della visita scegliendo la clinica, medico e la data della visita se tale prenotazione non era già presa da qualcun altro. Il paziente può essere registrato nel sistema senza creare l’account per esso, perché ad esempio può arrivare in clinica per la prima volta essendo in situazione grave, in tal caso potrà creare l’account più tardi.

In ogni clinica ci sono i suoi lavoratori, qualcuno di essi sono medici oppure amministratori. I lavoratori hanno il codice fiscale, nome, cognome, età e orario di lavoro. Non tutti i lavoratori hanno bisogno di avere l’account per sé perché non hanno bisogno di accedere sul sito, ma la clinica dovrà comunque tenere traccia di loro. Soltanto medici e amministratori devono avere l’account se vogliono disporre le prenotazioni oppure cambiare i dati del sistema. Ogni medico può lavorare in più cliniche mentre altro personale lavora solo in una.

Il medico ha la specializzazione. Tal orario può essere cambiato solo da amministratore di quella clinica.

Amministratore è definito dal livello di amministrazione. Il livello più basso può soltanto modificare i dati relativi alle prenotazioni dentro la sua clinica, mentre quello con livello più alto può cambiare qualsiasi dati.

Le cliniche sono caratterizzate da: un luogo, un orario, e dai reparti. Nella clinica si può tenere traccia dei dipendenti presenti e quelli passati, dei quali si sa anche la data di licenziamento. Nella storia di impiego possono essere dipendenti quelli che erano licenziati prima, ma poi sono tornati a lavorare nella stessa clinica. Ogni lavoratore corrente ha lo stipendio e può essere assegnato a un dipartimento.

Ogni reparto ha un medico primario, la specializzazione e la lista di lavoratori. I medici che hanno la specializzazione specifica possono lavorare solo nel reparto di questa specializzazione e la clinica non può avere multipli dipartimenti con la stessa specializzazione. Solo medico primario può lavorare nel dipartimento con la specializzazione diversa. Altri lavoratori non avendo il vincolo sulla specializzazione possono lavorare nei dipartimenti diversi, ma nella stessa clinica.

Il sistema deve tenere informazioni sugli esiti che sono rappresentati come un documento e sono associati con il medico e la clinica specifica dove era fatto quell’esito.

Tutti i pazienti aventi l’account possono fare le prenotazioni per le visite ai medici in qualche clinica desiderata. Tali prenotazioni sono gestite sia dai medici che da amministratori di clinica dove lavorano.

L’orario di lavoro sia della clinica che dal dottore è definito con il tempo di inizio e fine per un giorno della settimana oppure per una data specifica.

La specializzazione di un dipartimento o di un dottore è determinata dal nome.

# Progettazione concettuale

## Glossario dei termini

Prima nella richiesta abbiamo specificato:

* I termini evidenziati in rosso rappresentano lo stesso concetto
* I termini evidenziati in blu sono ambigui ed imprecisi

Termini ambigui ed imprecisi:

Il luogo della clinica è rappresentato dalla via, città, nazione e cap.

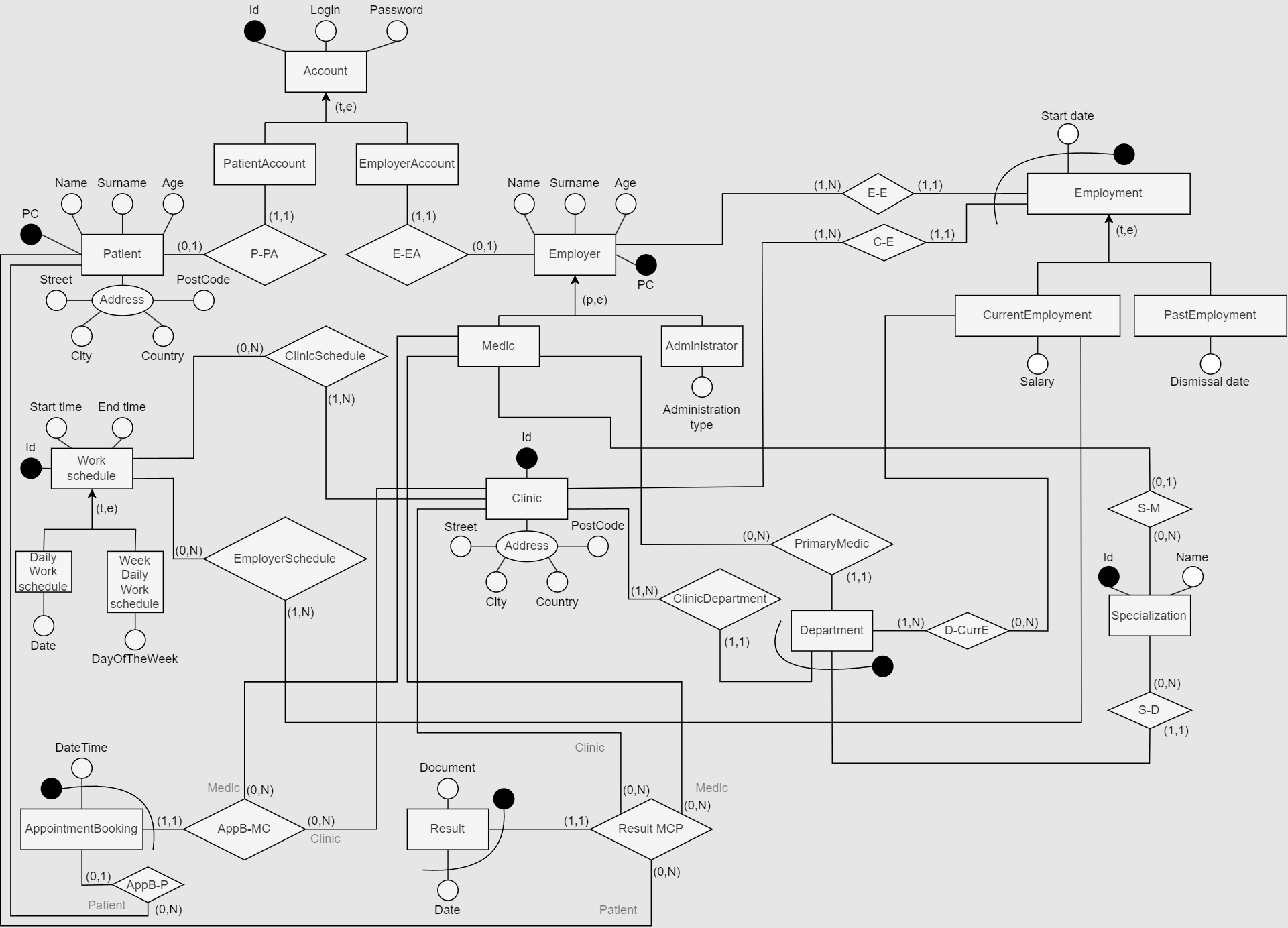
|  |  |  |  |
| --- | --- | --- | --- |
| Termine  (Nome in DB) | Descrizione | Sinonimi | Collegamenti |
| Clinica  (Clinic) | Sede per i dipendenti |  | Dipendente, Dipartimento, Esito, Prenotazione della visita, Orario di lavoro |
| Dipendente  (Employer) | Lavoratore della clinica | Lavoratore | Medico, Amministratore, Clinica, Account, Orario di lavoro |
| Medico  (Medic) | Medico della clinica | Dottore | Dipendente, Esito, Prenotazione della visita, Dipartimento, Clinica, Specializzazione |
| Amministratore  (Administrator) | Gestore del sito e dati del sistema |  | Dipendente, Clinica, Dipartimento |
| Dipartimento  (Department) | Sezione specifica della clinica | Reparto | Dipendente, Medico, Amministratore, Clinica, Specializzazione |
| Specializzazione  (Specialization) | Caratteristica che distingue medici e dipartimenti |  | Medico, Dipartimento |
| Paziente  (Patient) | Paziente delle cliniche |  | Account, Esito, Prenotazione della visita |
| Account  (Account) | Strumento per accedere al sito |  | Paziente, Dipendente |
| Orario di lavoro  (Work Schedule) | Periodo di attività di dipendenti e clinica |  | Clinica, Dipendente |
| Prenotazione della visita  (Appointment booking) | Meccanismo per far visitare i pazienti ai medici |  | Clinica, Medico, Paziente |
| Esito  (Result) | Esito di paziente che era dato da un medico nell’una clinica |  | Clinica, Medico, Paziente |

## Costruzione dello schema concettuale

Alcune osservazioni sulla costruzione dello schema ER:

* Per rappresentare l’impiego di lavoratori non basta avere l’associazione tra la clinica e il lavoratore, perché vogliamo rappresentare anche la possibilità del riassunzione del lavoratore che era stato licenziato. Cioè dobbiamo introdurre un’altra entità “Impiego (Employment)”.
* Dobbiamo introdurre 4 generalizzazioni:
  + Per gli account di pazienti e dipendenti (t, e)
  + Per i dipendenti medici, amministratori e altri (p, e)
  + Per l’orario di lavoro per il giorno della settimana e una data specifica (t, e)
  + Per l’impiego corrente e passato (t, e)

Nella seguente pagina è riportato lo schema ER iniziale



Dizionario dei dati (entità)

|  |  |  |  |
| --- | --- | --- | --- |
| Entità | Descrizione | Attributi | Identificatore |
| Account | Account del sito | ID, Login, Password | ID |
| PatientAccount | Account del paziente | Gli stessi di Account | Gli stessi di Account |
| EmployerAccount | Account del dipendente | Gli stessi di Account | Gli stessi di Account |
| Patient | Paziente | PC, Name, Surname, BirthDate, Address | PC |
| Employer | Dipendente | PC, Name, Surname, BirthDate | PC |
| Specialization | Caratteristica che distingue medici e reparti | ID, Name | ID |
| Medic | Medico | Gli stessi di Employer | Gli stessi di Employer |
| Administrator | Amministratore del sito | Administration type, Gli stessi di Employer | Gli stessi di Employer |
| Clinic | Luogo per lo svolgimento delle visite | ID, Address | ID |
| Employment | Impiego del dipendente | Start date | Start date, Employer, Clinic |
| CurrentEmployment | Impiego attuale | Salary, Gli stessi di Employment | Gli stessi di Employment |
| PastEmployment | Impiego passato | Dismissal date, Gli stessi di Employment | Gli stessi di Employment |
| Department | Dipartimento della clinica |  | Clinic, Specialization |
| WorkSchedule | Orario di lavoro | ID, Start time, End time | ID |
| DailyWorkSchedule | Orario di lavoro per una data specifica | Date, Gli stessi di WorkSchedule | Gli stessi di WorkSchedule |
| WeekDailyWorkSchedule | Orario di lavoro per un giorno della settimana | Day of the week, Gli stessi di WorkSchedule | Gli stessi di WorkSchedule |
| Result | Esito della visita | Date, Document | Date, Clinic, Medic, Patient |
| AppointmentBooking | Prenotazione dell’una visita | Date time | Date time, Medic, Clinic |

Dizionario dei dati (relationship)

|  |  |  |
| --- | --- | --- |
| Relationship | Descrizione | Componenti |
| P-PA | Collegamento tra un paziente e il suo account | Patient, PatientAccount |
| E-EA | Collegamento tra un dipendente e il suo account | Employer, EmployerAccount |
| E-E | Rapporto tra il dipendente e lo stato di lavoro | Employer, Employment |
| C-E | Rapporto tra la clinica e il lavoro | Clinic, Employment |
| EmployerSchedule | Orario dei dipendenti | WorkSchedule, CurrentEmployment |
| ClinicSchedule | Orario della clinica | WorkSchedule, Clinic |
| PrimaryMedic | Scelta del medico primario | Medic, Department |
| ClinicDepartment | Divisione della clinica in dipartimenti | Clinic, Department |
| D-CurrE | Dipendenti del reparto | Department, CurrentEmplyment |
| S-M | Divisione dei medici in base alla specializazione | Specialization, Medic |
| S-D | Divisione dei repati in base alla specializazione | Specialization, Department |
| AppB-MC | Gestione degli appuntamenti in base alla clinica e al medico | AppointmentBooking, Medic, Clinic |
| Result MCP | Collegamneto tra il risultato e le sue parti consone | Result, Medic, Clinic, Patient |

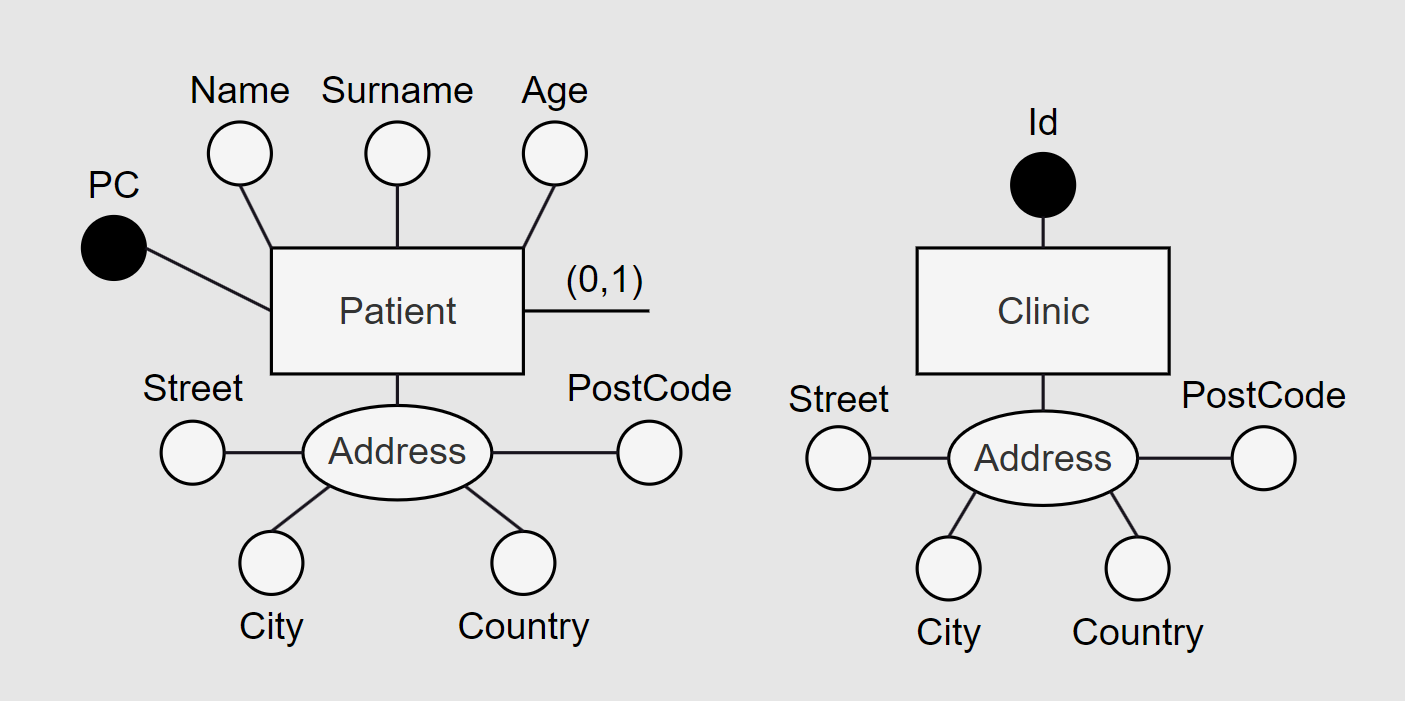
Vincoli non esprimibili:

1. I medici devono avere la stessa specializzazione del reparto in cui lavorano se hanno la specializzazione
2. I medici primari devono lavorare nella stessa clinica dove sta il suo dipartimento

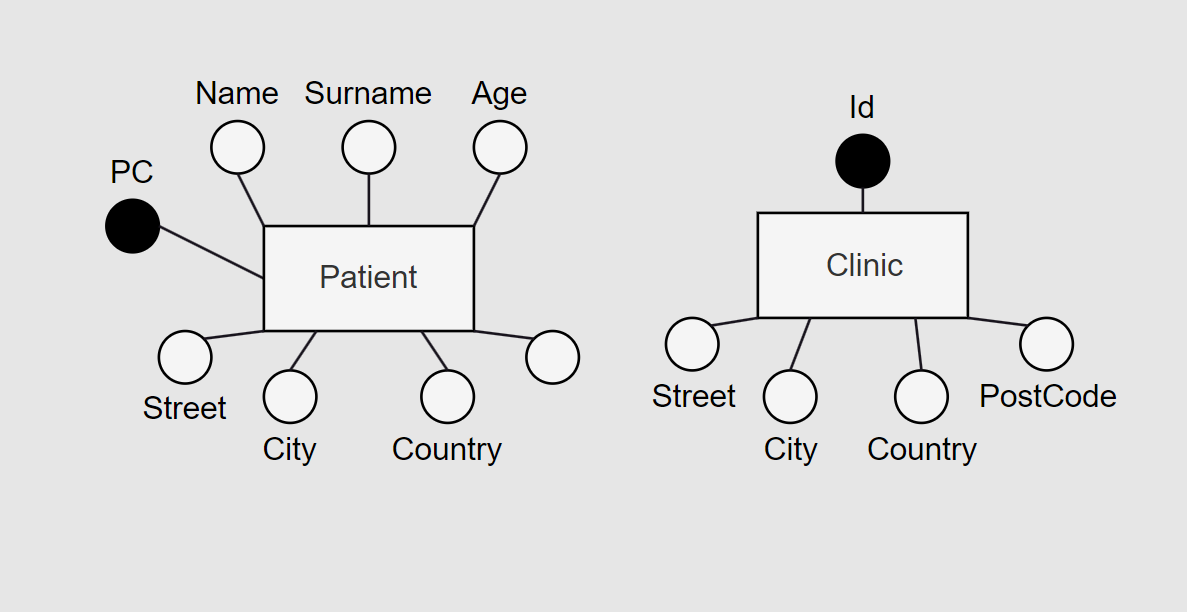
# Progettazione logica

## Eliminazione di attributi composti

Attributi composti nel paziente e clinica:

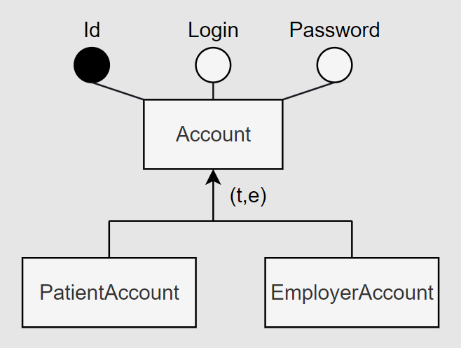


Eliminazione:

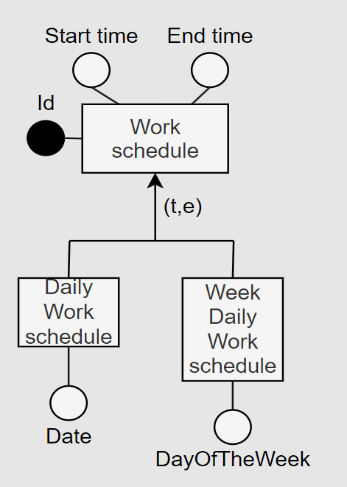


## Eliminazione delle gerarchie

Prima di eliminare le generalizzazioni dobbiamo considera le operazioni che vogliamo fare sulle entità coinvolte.

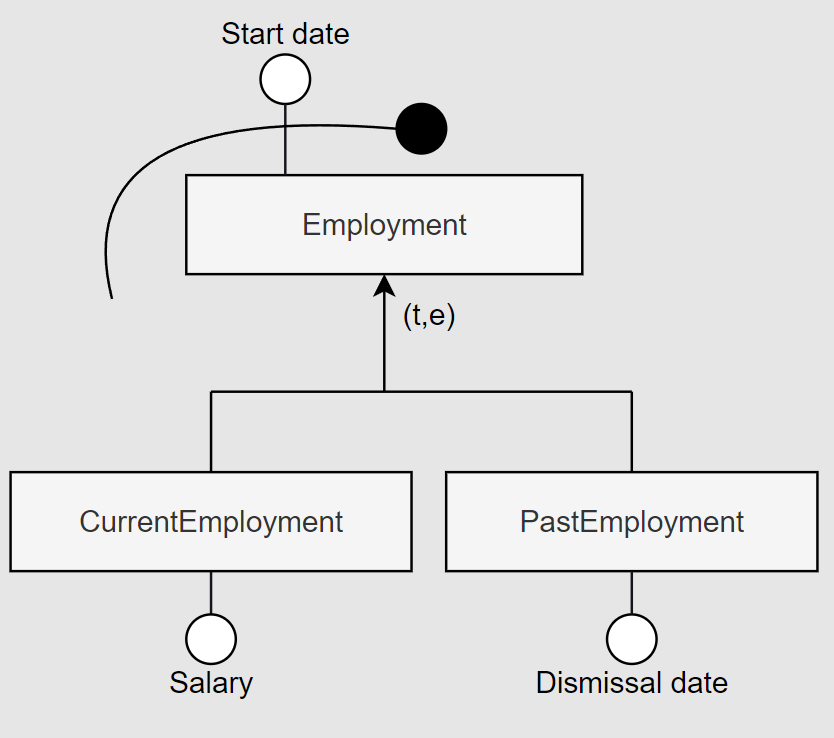
L’account viene letto da DB solo nella fase di autenticazione. Puoi avendo l’ID dell’account siamo interessati all’autorizzazione che determina il funzionamento del sito rispetto ai tipi di account.

Per eliminare tale generalizzazione ci conviene di accorpare i figli al padre aggiungendo il tipo di figlio.

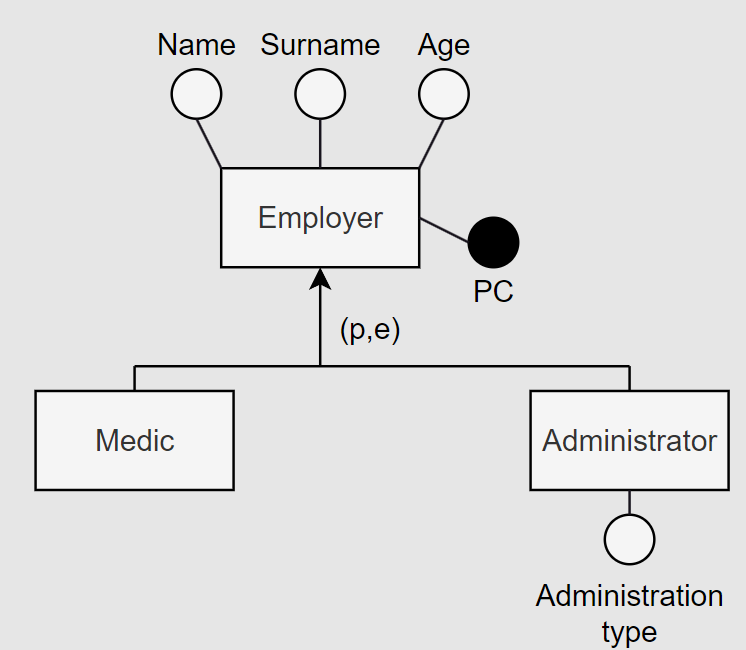


L’orario sia del medico che della clinica è disposto sul sito. Quello settimanale si può trovare accedendo nella pagina opposta della clinica. Quello orario di data specifica viene mostrato soltanto per il mese corrente ed e disponibile sulla stessa pagina.

Avendo l’operazione che richiede di mostrare entrambi tipi di orario ci conviene di accorpare i figli al padre, come abbiamo fatto con l’account.

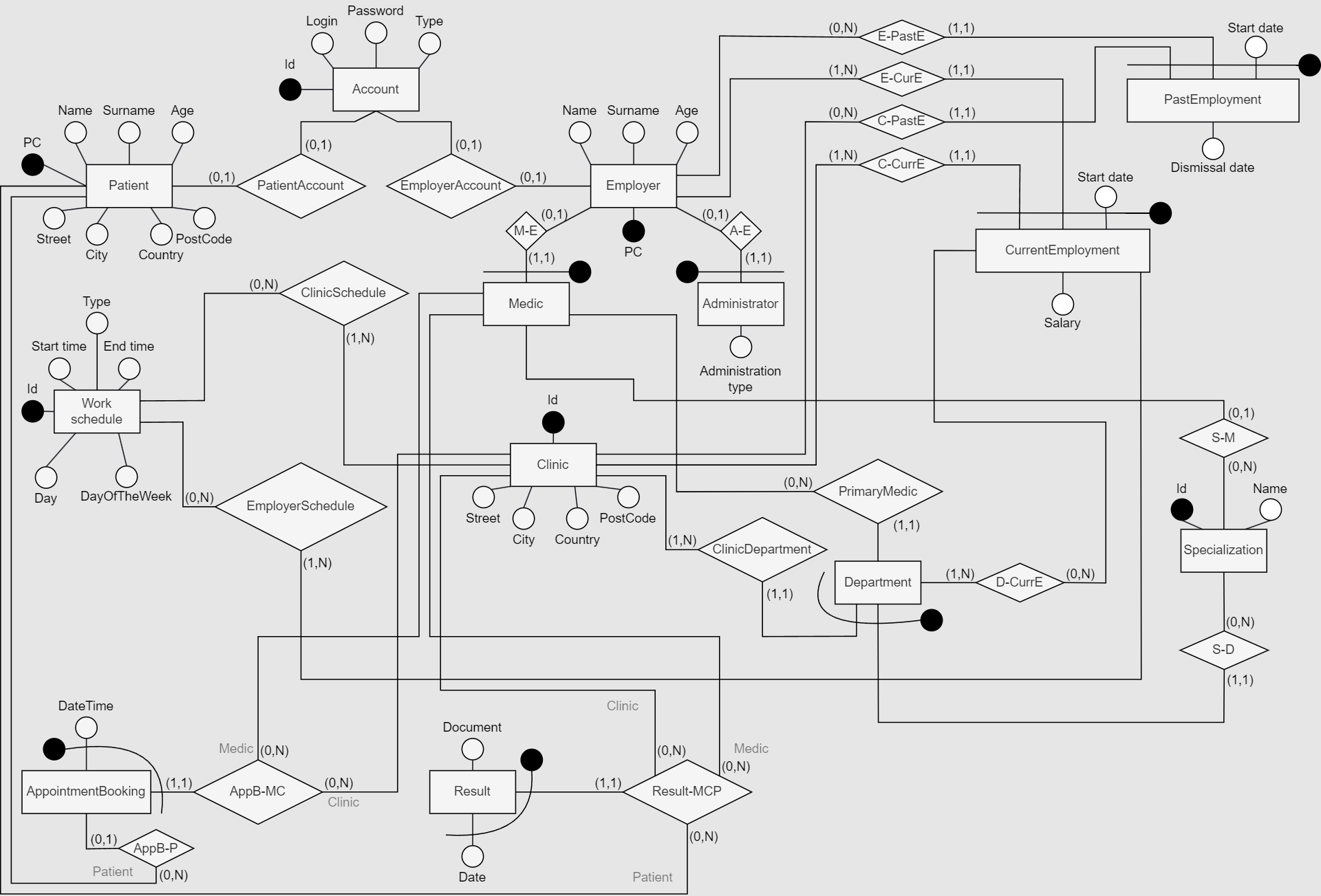
L’impiego viene letto soltanto dagli amministratori del sistema. Loro hanno due pagine del sito per leggere impieghi della clinica. Una pagina per impieghi correnti e l’altra per impieghi passati. Possono accedere anche alla storia dell’impiego di un lavoratore della clinica.

Siccome l’accesso è separato e viene fatto soltanto alle figlie ci conviene eliminare questa generalizzazione accorpando il genitore alle figlie.

 Riguardando l’accesso sui dipendenti possiamo stabilire che siamo interessati a determinare qualsiasi entità coinvolta nella generalizzazione perché gli amministratori vogliono sapere le informazioni riguardando tutti i tipi di dipendenti e i pazienti invece sono interessati a determinare i medici.

Essendo anche la generalizzazione parziale il modo migliore per eliminarla é di sostituirla con le associazioni tra figlio e padre.

Nella seguente pagina è riportato lo schema ER finale



## Traduzione verso il modello relazionale

|  |
| --- |
| **Account** (Id, Login, Password, Type)  **Clinic** (Id, Street, City, Country, PostCode)  **Specialization** (Id, Name)  **Patient** (PC, Name, Surname, BirthDate, Street, City, Country, PostCode, *AccountId*\*)  Patient.AccountId → Account.Id  **Employer** (PC, Name, Surname, BirthDate, *AccountId*\*)  Employer.AccountId → Account.Id  **Medic** (*EmployerPC*, *SpecializationId\**)  Medic.EmployerPC → Employer.PC  Medic.SpecializationId → Specialization.Id  **Administrator** (*EmployerPC*, AdministrationType)  Administrator.EmployerPC → Employer.PC  **CurrentEmployment** (*EmployerPC*, *ClinicId,* StartDate, Salary)  CurrentEmployment.EmployerPC → Employer.PC  CurrentEmployment.ClinicId → Clinic.Id  **PastEmployment** (*EmployerPC*, *ClinicId*, StartDate, DismissalDate)  PastEmployment.EmployerPC → Employer.PC  PastEmployment.ClinicId → Clinic.Id  **Department** (*ClinicId*, *SpecializationId*, *PrimaryMedicEmployerPC*)  Department.ClinicId → Clinic.Id  Department.SpecializationId → Specialization.Id  Department.PrimaryMedicEmployerPC → Medic.EmployerPC  **WorkSchedule** (Id, StartTime, EndTime, Day\*, DayOfTheWeek\*, Type)  **Result** (Date, *ClinicId*, *MedicEmployerPC*, *PatientPC*, Document)  Result.ClinicId → Clinic.Id  Result.MedicEmployerPC → Medic.EmployerPC  Result.PatientPC → Patient.PC  **AppointmentBooking** (DateTime, ClinicId, MedicEmployerPC, PatientPC\*)  AppointmentBooking.ClinicId → Clinic.Id  AppointmentBooking.MedicEmployerPC → Medic.EmployerPC  AppointmentBooking.PatientPC → Patient.PC  **EmployerSchedule** (*CurrEmployerPC*, *CurrEmployerStartDate*, *CurrEmployerClinicId*, *WorkScheduleId*)  EmployerSchedule. CurrEmployerPC → CurrentEmployment.EmployerPC  EmployerSchedule. CurrEmployerStartDate → CurrentEmployment.StartDate  EmployerSchedule. CurrEmployerClinicId → CurrentEmployment.ClinicId  EmployerSchedule. WorkScheduleId → WorkSchedule.Id  **ClinicSchedule** (*ClinicId*, *WorkScheduleId*)  ClinicSchedule.ClinicId → Clinic.Id  ClinicSchedule.WorkScheduleId → WorkSchedule.Id  **DepartmentCurrentEmployment** (*CEEmployerPC*, *CEStartDate*, *ClinicId*, *DepSpecializationId*)  DepartmentCurrentEmployment.CEEmployerPC → CurrentEmployment.EmployerPC  DepartmentCurrentEmployment.CEStartDate → CurrentEmployment.StartDate  DepartmentCurrentEmployment.ClinicId → CurrentEmployment.ClinicId  DepartmentCurrentEmployment.ClinicId → Department.ClinicId  DepartmentCurrentEmployment.DepSpecializationId → Department.SpecializationId |

Attributi con asterisco possono essere nulli. Attributi sottolineati rappresentano le chiave primarie, mentre quelle in corsivo rappresentano le chiave referenziali.

Avendo scelto dentro il *DepartmentCurrentEmployment* un solo unico *ClinicId* siamo riusciti a rappresentare il vincolo che in dipartimento devono lavorare medici della stessa clinica.

# Progettazione fisica

## Struttura del file assegnati al progetto

Abbiamo allegato i seguenti tipi di file insieme a questa relazione:

* File di codice *MySQL* per la creazione di tabelle, procedure, trigger, viste e popolamento di basi di dati tramite istruzioni *DDL*.
* File di codice *MySQL* per il popolamento di tabelle tramite file .csv allegati.
* Elenco di file .csv per tutti le tabelle. Questi file si usano per l’import dentro le basi di dati.

## Tabelle e viste

Il popolamento delle tabelle viene effettuato tramite il codice oppure il file .*csv*.

È stata introdotta una vista contenente le informazioni di cliniche italiane.

## Procedure

Sono state implementate tre procedure e una funzione:

1. *CancelAppointmentBooking* per poter cancellare la prenotazione di un paziente per un medico di una clinica e data desiderati. Dopo di aver trovato la prenotazione necessaria imposta il riferimento nullo al paziente.
2. *ListClinicWithTheSumOfPaidSalaries* che elenca lo stipendio pagato da tutte le cliniche.
3. *ListWorkScheduleOfTheClinic* elenca invece l’orario della clinica visualizzando anche l’orario per un giorno specifico entro i prossimi giorni desiderati.
4. *GetClinicSumOfThePaidSalaries* è una funzione che restituisce lo stipendio pagato da una clinica specifica.

## Trigger

Sono introdotte anche 6 trigger per poter mantenere alcuni vincoli non esprimibili:

1. Due trigger che servono per poter esprimere la generalizzazione totale di *WorkSchedule*. Iniziano a lavorare prima degli inserimenti e degli aggiornamenti dei dati. Servono per assegnare il valore nullo alla data se il tipo di *WorkSchedule* è “*WeekDailyWorkSchedule*”, e altrimenti assegnare valore nullo al giorno della settimana.
2. Due trigger che servono per esprimere il vincolo che il medico primario deve lavorare nella stessa clinica del suo dipartimento. Se viene inserito un dipartimento con il medico che non lavora nella stessa clinica viene mandato un segnale di errore in MySQL. Se invece questo succede ad un aggiornamento allora dobbiamo soltanto lasciare il medico primario vecchio.
3. Un trigger che serve per il controllo del medico e la clinica mentre si inserisce un esito. Quel trigger controlla che il medico lavori in questa clinica.
4. Un trigger che serve per il controllo del medico e la clinica mentre si inserisce una prenotazione della visita. Quel trigger controlla che il medico lavori in questa clinica.