

**Progetto di Basi di Dati
A.A. 2015/2016**

Daniel De Gaspari, matricola n° 1078058 ddegaspa
Giulia Petenazzi, matricola n° 1093298 Username: gpetenaz

Nome progetto: **YouTooth**



Il progetto e' stato caricato sull'account ddegaspa

Dati d'accesso per l'interfaccia web:

- Privilegi di Medico: Username: Daniel Password: Daniel
- Privilegi di Segretaria: Username: Giulia Password: Giulia
- Privilegi di Paziente: Username: mario.legionario Password: password_mario

Indice:

1) Abstract.....	5
2) Acquisizione e analisi dei requisiti.....	5
2.1 Descrizione dei requisiti.....	5
2.2 Specifiche sulle operazioni.....	6
2.3 Glossario dei termini.....	7
2.4 Strutturazione dei requisiti.....	8
3) Progettazione concettuale.....	10
3.1 Descrizione testuale delle classi.....	10
3.1.1 Persona.....	10
3.1.2 Personale.....	10
3.1.3 Medico.....	10
3.1.4 Medico attivo.....	10
3.1.5 Medico licenziato.....	10
3.1.6 Segretaria.....	10
3.1.7 Segretaria attiva.....	10
3.1.8 Segretaria licenziata.....	11
3.1.9 Paziente.....	11
3.1.10 Account.....	11
3.1.11 Visita.....	11
3.1.12 Sezione Visita.....	11
3.1.13 TipoVisita.....	11
3.2 Descrizione testuale delle Associazioni.....	12
3.2.1 Account - Paziente: “Possesso”.....	12
3.2.2 Account- Segretaria attiva: “Possesso”.....	12
3.2.3 Account – Medico attivo: “Prenotazione”.....	12
3.2.4 Paziente - Visita: “Prenotazione”.....	12
3.2.5 Medico - TipoVisita: “Specializzazione”.....	12
3.2.6 Visita - SezioniVisita: “Divisione”.....	12
3.2.7 SezioniVisita – TipoVisita: “Appartenenza”.....	12
3.2.8 Medico – Sezione Visita: “Effettuazione”.....	12
3.3 Descrizione delle generalizzazioni.....	13
3.3.1 Persona: Paziente, Personale.....	13
3.3.2 Personale: Segretaria, Medico.....	13
3.3.3 Medico: Medico attivo, Medico licenziato.....	13
3.3.4 Segretaria: Segretaria attiva, Segretaria licenziata.....	13
3.4 Descrizione dello schema concettuale in forma grafica: lo schema E-R.....	14
3.5 Dizionario dei dati: entità.....	14
3.6 Dizionario dei dati: relationship.....	15
3.7 Vincoli non esprimibili dallo schema E-R.....	16

7) View, Funzioni, Procedure, Triggers.....	30
7.1 Funzioni.....	30
7.1.1. is_operazione_disponibile.....	30
7.1.2. prenotazioni_future.....	31
7.1.3. totali.....	31
7.2 Procedure.....	32
7.2.1. ri_assumi.....	32
7.2.2. assumi.....	33
7.2.3. disponibile.....	33
7.3 Triggers.....	35
7.3.1. check_data_ora_insert.....	35
7.3.2. check_data_ora_update.....	35
7.3.3. segretaria_non_medico_insert.....	36
7.3.4. segretaria_non_medico_update.....	36
7.3.5. medico_non_segretaria_insert.....	36
7.3.6. medico_non_segretaria_update.....	37
7.3.7. check_prezzo_positivo_ins.....	37
7.3.8. check_prezzo_positivo_upd.....	37
7.3.9. check_medico_licenziato_libero_insert.....	38
7.3.10. check_medico_licenziato_libero_update.....	38
7.3.11. check_medico_libero_specializzato_insert.....	39
7.3.12. check_medico_libero_specializzato_update.....	40
8. Interfaccia WEB.....	41
8.1 Organizzazione delle pagine e operazioni.....	41
8.2 Dettagli autenticazione.....	44

1) Abstract

Il progetto “YouTooth” prevede la realizzazione di una base di dati per la gestione delle prenotazioni delle visite di un poliambulatorio odontoiatrico. In particolare, vengono gestiti i dati relativi ai medici che operano nello studio, ai tipi di prestazione in cui essi sono specializzati, e i dati relativi alla/alle segretarie che vi lavorano. Vengono inoltre memorizzati i dati relativi ai pazienti e alle loro rispettive visite. Ogni paziente, ogni medico e ogni segretaria dello studio possiede un proprio account che consentirà loro di accedere all'applicazione web che si interfaccia con la base di dati. Le operazioni tipiche sono la visualizzazione delle visite prenotate, sia dal punto di vista del paziente sia dal punto di vista del medico, la prenotazione di nuove visite relative a pazienti già registrati e la registrazione di nuovi pazienti.

2) Acquisizione e analisi dei requisiti

2.1 Descrizione dei requisiti

Il progetto “YouTooth” prevede la realizzazione di una base di dati per la gestione delle prenotazioni delle visite di un poliambulatorio odontoiatrico.

Nella realtà che vogliamo modellare vengono coinvolte persone aventi diversi ruoli: abbiamo i pazienti e il personale, diviso a sua volta tra segretarie e medici.

Dei medici che operano nello studio (circa 5), ci interessano: il nome, il cognome, il numero di telefono, la data di nascita, la residenza, il sesso e il codice fiscale (che lo identifica univocamente), la data di assunzione e l'eventuale data di licenziamento, e l'indirizzo e-mail. I dentisti possono effettuare un certo set di prestazioni (ad esempio, pulizia dei denti, carie, otturazioni, ecc e in media un medico è specializzato in sette tipologie di visita). Mediamente ogni visita appartiene a due tipologie di visita.

Delle segretarie che operano nello studio (1 al momento), ci interessano: il nome, il cognome, il numero di telefono, la sua data di nascita, la residenza, il sesso, la data di assunzione e l'eventuale data di licenziamento, il codice fiscale (che la identifica univocamente), e l'indirizzo e-mail.

Dei pazienti dello studio dentistico (circa 1000), ci interessano il nome, il cognome, il numero di telefono, la data di nascita, la residenza, il sesso e il codice fiscale (che lo identifica univocamente), oltre all'indirizzo e-mail.

Ogni soggetto coinvolto nella realtà dello studio, ovvero medici, segretarie e pazienti, avrà in dotazione un proprio account.

Si vuole poter gestire inoltre il caso in cui un medico (o una segretaria) sia anche paziente dello studio: questi soggetti saranno dotati di due account: uno da medico(o segretaria) e uno da paziente.

Ogni paziente, grazie all'account datogli in dotazione, sarà in grado di accedere all'applicazione web che si interfaccia con la base di dati: l'obiettivo è quello di fornire ai pazienti dello studio uno strumento che consenta loro di visionare i dati relativi allo storico delle proprie visite, e gli eventuali appuntamenti futuri, oltre a poter gestire i propri dati personali.

L'account del medico consentirà al medico stesso di accedere all'applicazione web che si interfaccia con la base di dati: tramite l'account, i medici potranno gestire i propri dati personali, visionare gli appuntamenti passati e futuri che li vedono coinvolti, e registrare nuovo personale, sia medici, sia segretarie.

Anche la segretaria possiede un account, mediante il quale può accedere all'applicazione web che si interfaccia con la base di dati e che consentirà la normale gestione dello studio dentistico. Le operazioni che può compiere sono: gestione dei propri dati personali, registrazione di nuovo personale (medici e segretarie), prenotazione di visite per i clienti già registrati, modifica dei dati relativi ad una visita già prenotata, oltre all'aggiunta di tipologie di visite offerte dallo studio, alla specializzazione di medici già presenti in tipologie di visite presenti, e al licenziamento di personale.

Ogni account sarà caratterizzato da uno username, da una password, e dai privilegi offerti dall'account a seconda della tipologia di persona (paziente, medico o segretaria) a cui tale account appartiene. Gli account saranno forniti dalla segretaria in sede di registrazione nello studio (nel caso di pazienti) o in sede di assunzione (in caso di medici o segretarie). I pazienti avranno inoltre la

possibilità di iscriversi allo studio in modo autonomo, ottenendo così anche un account personale, tramite interfaccia web.

Per motivi legislativi / statistici, si terrà traccia del personale passato dello studio, specificando la data del licenziamento del medico/segretaria. Al momento del licenziamento, il soggetto passivo di tale azione perderà il proprio account di medico/segretaria, mantenendo però l'eventuale account da paziente.

Lo studio offre varie tipologie di operazioni effettuate nelle visite, identificate dal nome della prestazione; ogni tipo di prestazione è inoltre caratterizzata da un prezzo e si vuole avere la possibilità di memorizzarne una breve descrizione.

Le visite quindi possono essere divise in più sezioni e ogni sezione della visita appartiene a una sola tipologia di visita e deve essere effettuata da almeno un medico, o possibilmente da più medici, specializzati in quella tipologia di visita. (Ad esempio la visita “x” può essere divisa in “pulizia dei denti” e “otturazione carie” e la sezione “visita x – otturazione carie” può essere effettuata da almeno un medico o da più medici specializzati in “otturazione carie”).

Le visite (circa 50 al giorno), saranno caratterizzate da una data, un orario e un certo prezzo che il paziente deve sostenere per una determinata visita, che sarà dato dalla somma delle varie prestazioni singole appartenenti alla visita. Un paziente può sostenere una sola visita in una determinata data ad un determinato orario; allo stesso modo un medico può sostenere una sola visita in una determinata data ad un determinato orario.

Lo studio rimane aperto dal lunedì al venerdì dalle 9 alle 18.

2.2 Specifiche sulle operazioni

Per la nostra applicazione, riportiamo di seguito le operazioni previste, corredate da una breve descrizione e dal carico operativo previsto (= previsione sul numero di volte per cui si ricorre all'utilizzo di queste operazioni).

- Operazione 1: registrazione di un nuovo paziente, indicandone tutti i suoi dati.
 - Operazione da effettuare in media 2 volte al giorno.
- Operazione 2: registrazione di un nuovo medico/segretaria, indicandone tutti i suoi dati.
 - Operazione da effettuare in media 2 volte all'anno.
- Operazione 3: inserisci una nuova tipologia di visita.
 - Operazione da effettuare in media 2 volte all'anno.
- Operazione 4: specializzazione di un medico ad una tipologia di visita.
 - Operazione da effettuare in media 3 volte all'anno.
- Operazione 5: per un certo paziente mostrare i dati relativi alle visite prenotate, passate e future, con indicazione della spesa totale sostenuta per ognuna di queste visite.
 - Operazione da effettuare in media 30 volte al giorno.
- Operazione 6: stampa delle informazioni personali dei soggetti presenti nello studio.
 - Operazione da effettuare in media 50 volte al giorno.
- Operazione 7: per un certo medico, trova i pazienti che sono stati visitati o che saranno visitati da tale medico.
 - Operazione da effettuare in media 1 volta al giorno.
- Operazione 8: prenotazione di una nuova visita.
 - Operazione da effettuare in media 50 volte al giorno.
- Operazione 9: licenziamento di personale (medico o segretaria).
 - Operazione da effettuare in media 2 volte all'anno.
- Operazione 10: modifica dei dati relativi ad una visita.
 - Operazione da effettuare in media 5 volte al giorno.
- Query1: Trovare le due tipologie di visite meno richieste nell' ultimo anno, mostrando anche il numero di volte in cui sono state richieste.
 - Operazione da effettuare in media 1 volta al mese.
- Query2: Trovare i clienti che hanno fatto almeno una visita la cui tipologia è di prezzo più costoso.
 - Operazione da effettuare in media 1 volta al mese.

- Query3: Trovare i clienti che hanno fatto almeno una visita, il cui prezzo è il secondo più costoso.
- Operazione da effettuare in media 1 volta al mese.
- Query4: Mostrare i medici e le segretarie che sono anche pazienti dello studio e che non si fanno visitare da più di 5 anni.
- Operazione da effettuare in media 1 volta al mese.
- Query5: Mostra i medici che hanno effettuato visite a pazienti che hanno fatto SOLO estrazioni di denti del giudizio.
- Operazione da effettuare in media 1 volta al mese.
- Query6: Mostra la spesa totale di ogni cliente, visualizzando cognome e nome del cliente, il codice fiscale e la sua spesa, ordinati per cognome e nome.
- Operazione da effettuare in media 1 volta al mese.

2.3 Glossario dei termini:

Termine	Descrizione	Sinonimi	Collegamenti
Persona	Rappresenta i soggetti che partecipano alla vita dello studio dentistico, ovvero pazienti, medici e segretarie.		
Paziente	Paziente dello studio dentistico.		Visita Account
Personale	Personale dello studio dentistico, possono essere medici o segretarie		
Medico	Medico che lavora o che ha lavorato nello studio dentistico	Dentista	TipoVisita
Segretaria	Segretaria che lavora o che ha lavorato nello studio dentistico.		
Account	Indica un account, dotato di privilegi, assegnato a pazienti/medici/segretarie per l'accesso a determinate porzioni della base di dati.		Paziente, Segretaria, Medico
Visita	Visita prenotata nello studio dentistico	Prestazione Visita prenotata	SezioneVisita
SezioneVisita	Sezioni in cui è divisa una visita		Visita
TipoVisita	Tipologia della visita prenotata nello studio dentistico	Set di prestazioni Tipologia di visita	SezioneVisita Medico

2.4 Strutturazione dei requisiti

Frase di carattere generale:

Il progetto “YouTooth” prevede la realizzazione di una base di dati per la gestione delle prenotazioni delle visite di un poliambulatorio odontoiatrico.

Frase relative alle persone:

Nella realtà che vogliamo modellare vengono coinvolte persone aventi diversi ruoli: abbiamo i pazienti e il personale, diviso a sua volta tra segretarie e medici.

Frase relative ai pazienti:

Dei pazienti dello studio dentistico (circa 1000), ci interessano il nome, il cognome, il numero di telefono, la data di nascita, la residenza, il sesso e il codice fiscale (che lo identifica univocamente), oltre all'indirizzo e-mail.

Frase relative al personale:

Nella realtà che vogliamo modellare vengono coinvolte persone aventi diversi ruoli: abbiamo i pazienti e il personale, diviso a sua volta tra segretarie e medici.

Frase relative alle segretarie:

Delle segretarie che operano nello studio (1 al momento), ci interessano: il nome, il cognome, il numero di telefono, la sua data di nascita, la sua residenza, il sesso, la data di assunzione e l'eventuale data di licenziamento, il codice fiscale (che lo identifica univocamente), e l'indirizzo e-mail.

Frase relative ai medici:

Dei medici che operano nello studio (circa 5), ci interessano: il nome, il cognome, il numero di telefono, la data di nascita, la residenza, il sesso e il codice fiscale (che lo identifica univocamente), la data di assunzione e l'eventuale data di licenziamento, e l'indirizzo e-mail. I dentisti possono effettuare un certo set di prestazioni (ad esempio, pulizia dei denti, carie, otturazioni, ecc).

Frase relative agli account:

Ogni soggetto coinvolto nella realtà dello studio, ovvero medici, segretarie e pazienti, avrà in dotazione un proprio account.

Si vuole poter gestire inoltre il caso in cui un medico (o una segretaria) sia anche paziente dello studio: questi soggetti saranno dotati di due account: uno da medico(o segretaria) e uno da paziente. Ogni paziente, grazie all'account datogli in dotazione, sarà in grado di accedere all'applicazione web che si interfaccia con la base di dati: l'obiettivo è quello di fornire ai pazienti dello studio, uno strumento che consenta loro di visionare i dati relativi allo storico delle proprie visite, e gli eventuali appuntamenti futuri, oltre a poter gestire i propri dati personali.

L'account del medico consentirà al medico stesso di accedere all'applicazione web che si interfaccia con la base di dati: tramite l'account, i medici potranno gestire i propri dati personali, visionare gli appuntamenti passati e futuri che li vedono coinvolti, e registrare nuovo personale, sia medici, sia segretarie.

Anche la segretaria possiede un account, mediante il quale può accedere all'applicazione web che si interfaccia con la base di dati, che consentirà la normale gestione dello studio dentistico. Le operazioni che possono compiere sono: gestione dei propri dati personali, registrazione di nuovo personale (medici e segretarie), prenotazione di visite per i clienti già registrati, modifica dei dati relativi ad una visita già prenotata, oltre all'aggiunta di tipologie di visite offerte dallo studio, alla specializzazione di medici già presenti in tipologie di visite presenti, e licenziamento di personale. Ogni account sarà caratterizzato da uno username, da una password, e dai privilegi offerti dall'account a seconda della tipologia di persona (paziente, medico o segretaria) a cui tale account appartiene. Gli account saranno forniti dalla segretaria in sede di registrazione nello studio (nel caso di pazienti) o in sede di assunzione (in caso di medici o segretarie). I pazienti avranno inoltre la possibilità di iscriversi allo studio in modo autonomo, ottenendo così anche un account personale, tramite interfaccia web.

Per motivi legislativi / statistici, si terrà traccia del personale passato dello studio, specificando la data del licenziamento del medico/segretaria. Al momento del licenziamento, il soggetto passivo di tale azione perderà il proprio account di medico/segretaria, mantenendo però l'eventuale account da paziente.

Frase relative alle visite:

Le visite (circa 100 al giorno), saranno caratterizzate da una data, un orario e un certo prezzo che il paziente deve sostenere per una determinata visita, che sarà dato dalla somma delle varie prestazioni singole appartenenti alla visita. Un paziente può sostenere una sola visita in una determinata data ad un determinato orario; allo stesso modo un medico può sostenere una sola visita in una determinata data ad un determinato orario.

Frase relative alle sezioni di visita:

Le visite quindi possono essere divise in più sezioni e ogni sezione della visita appartiene a una sola tipologia di visita e deve essere effettuata da almeno un medico, o possibilmente da più medici, specializzati in quella tipologia di visita. (Ad esempio la visita "x" può essere divisa in "pulizia dei denti" e "otturazione carie" e la sezione "visita x – otturazione carie" può essere effettuata da almeno un medico o da più medici specializzati in "otturazione carie").

Frase relative ai tipi di visita:

Lo studio offre varie tipologie di operazioni effettuate nelle visite, identificate dal nome della prestazione; ogni tipo di prestazione è inoltre caratterizzata da un prezzo e si vuole avere la possibilità di memorizzarne una breve descrizione.

3) Progettazione concettuale

3.1 Descrizione testuale delle classi

3.1.1 Persona

L'entità Persona raccoglie le informazioni anagrafiche caratterizzanti un particolare soggetto, che può essere un personale dello studio o un paziente.

- Attributi:
 - codice fiscale: char
 - nome: varchar
 - cognome: varchar
 - telefono: varchar
 - data_nascita: datetime
 - residenza
 - sesso: char
 - email: varchar

3.1.2 Personale

L'entità Personale, figlia di Persona, rappresenta un componente del personale dello studio dentistico, che può essere un medico o una segretaria.

- Attributi aggiuntivi rispetto a Persona:
 - data_assunzione: date

3.1.3 Medico

L'entità Medico, figlia di Personale, rappresenta un medico che opera nello studio. Non presenta attributi aggiuntivi rispetto a Personale.

3.1.4 Medico attivo

L'entità Medico attivo, figlia di Medico, rappresenta un medico che lavora attualmente nello studio. Non presenta attributi aggiuntivi rispetto a medico.

3.1.5 Medico Licenziato

L'entità Medico Licenziato, figlia di Medico, rappresenta un medico che ha lavorato nello studio e che è stato licenziato.

- Attributi aggiuntivi rispetto a Medico:
 - data_licenziamento: date

3.1.6 Segretaria

L'entità Segretaria, figlia di Personale, rappresenta una segretaria che lavora nello studio. Non presenta attributi aggiuntivi rispetto a Personale.

3.1.7 Segretaria attiva

L'entità segretaria attiva, figlia di segretaria, rappresenta una segretaria che lavora attualmente nello studio. Non presenta attributi aggiuntivi rispetto a medico.

3.1.8 Segretaria licenziata

L'entità Segretaria Licenziata, figlia di segretaria, rappresenta una segretaria che ha lavorato nello studio e che è stata licenziata.

- Attributi aggiuntivi rispetto a Segretaria:
 - data_licenziamento: date

3.1.9 Paziente

L'entità Paziente, figlia di Persona, rappresenta un generico paziente dello studio dentistico. Non presenta attributi aggiuntivi rispetto a Persona.

3.1.10 Account

L'entità account rappresenta un account dato in dotazione ad una persona, che può essere medico, segretaria o paziente.

- Attributi:
 - username: varchar
 - password: varchar
 - privilegi: varchar

3.1.11 Visita

L'entità Visita rappresenta le visite che sono state prenotate.

- Attributi:
 - id_visita: int
 - data: date
 - ora: time

3.1.12 Sezione visita

L'entità sezione visita rappresenta le sezioni in cui una visita è divisa, dove ogni sezione corrisponde ad una tipologia della visita all'interno di una determinata visita.

3.1.13 TipoVisita

L'entità TipoVisita rappresenta le tipologie di visite rese disponibili dallo studio dentistico.

- Attributi:
 - descrizione: varchar
 - prezzo: numeric
 - note: varchar

3.2 Descrizione testuale delle Associazioni

3.2.1 Account - Paziente: “Possesso”

Molteplicità 1:1

Ogni account è posseduto da un unico paziente; un paziente possiede uno e un solo account.

3.2.2 Account – Segretaria attiva: “Possesso”

Molteplicità 1:1

Ogni account è posseduto da un unica segretaria attiva; una segretaria possiede uno e un solo account.

3.2.3 Account – Medico attivo: “Possesso”

Molteplicità 1:1

Ogni account è posseduto da un unico medico attivo; un medico attivo possiede uno e un solo account.

3.2.4 Paziente - Visita: “Prenotazione”

Molteplicità M:1

Ogni paziente può aver prenotato una o più visite. Una visita è stata prenotata da uno e un solo paziente.

3.2.5 Medico - TipoVisita: “Specializzazione”

Molteplicità N:M

Un medico può essere specializzato in più tipologie di visite; ad una tipologia di visita possono essere specializzati più medici.

3.2.6 Visita - SezioniVisita: “Divisione”

Molteplicità 1:N

Ogni visita è divisa in una o più sezioni (ad esempio, visita x - sigillatura e visita x - pulizia dei denti). Una SezioneVisita fa parte di esattamente una sola visita.

3.2.7 SezioniVisita - TipoVisita: “Appartenenza”

Molteplicità 1:N

Ogni sezione di una visita appartiene a una e una sola tipologia di visita. Una tipologia di visita può appartenere da zero a più sezioni di visita. (possiamo dunque avere tipologie di visite che non sono mai state richieste).

3.2.8 Medico – Sezione visita: “Effettuazione”

Molteplicità N:M

Un medico può effettuare più sezioni di visite, mentre una sezione di visita viene effettuata da uno o più medici.

3.3 Descrizione delle generalizzazioni

3.3.1 Persona: Paziente, Personale,

Lo schema concettuale della base di dati prevede una generalizzazione, che ha come entità genitore Persona, e come entità figlie Paziente e Personale. La generalizzazione è totale, in quanto le due tipologie di persone sono sufficienti a rappresentare la realtà di interesse. La generalizzazione è sovrapposta in quanto un paziente può essere un anche componente del personale.

3.3.2 Personale: Segretaria, Medico

Il personale è diviso in segretarie e medici, quindi l'entità genitore è personale mentre le entità figlie sono segretaria e medico. La generalizzazione è totale, in quanto le due tipologie di persone sono sufficienti a rappresentare la realtà di interesse. La generalizzazione è esclusiva perché una segretaria non può essere anche un medico.

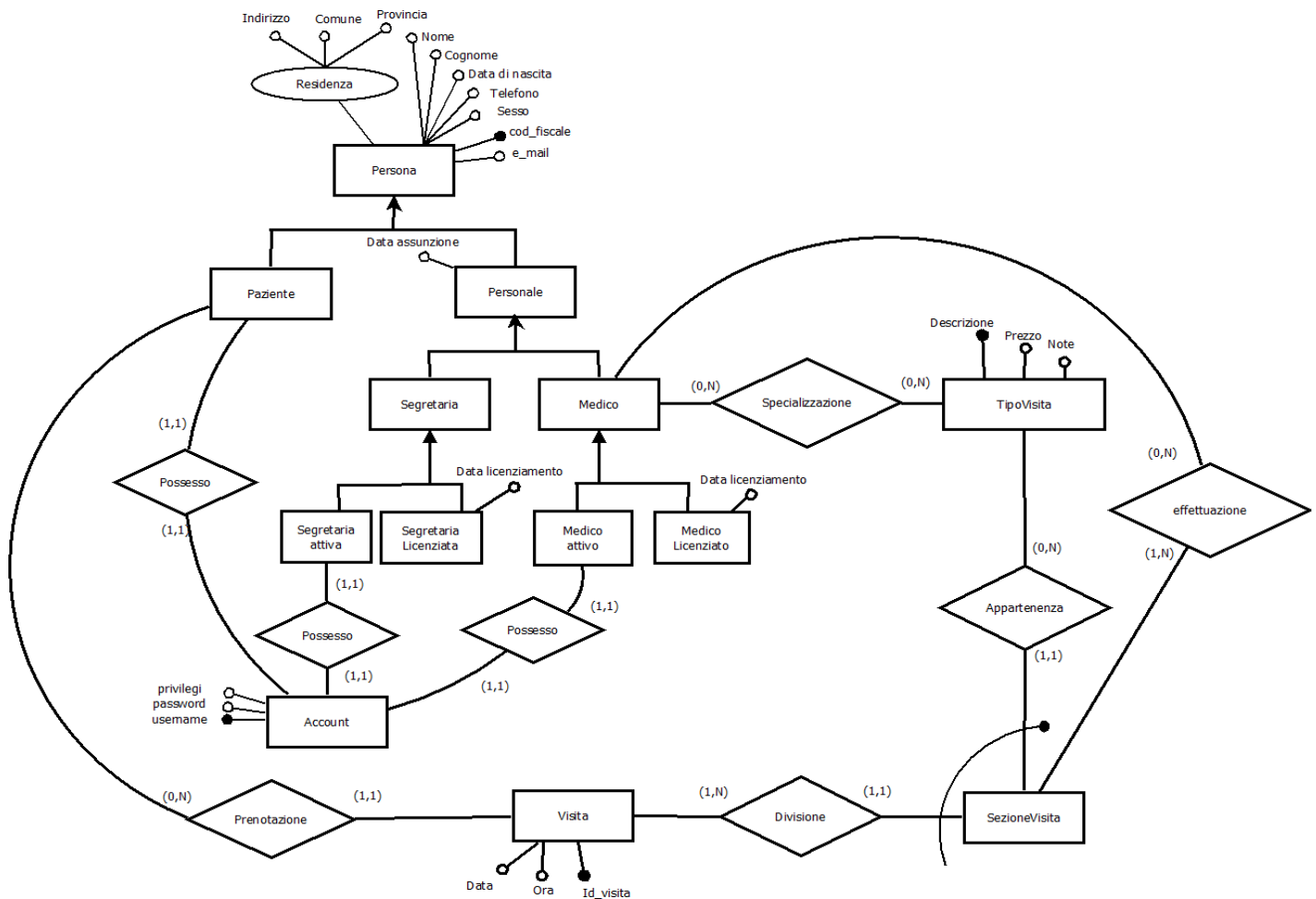
3.3.3 Medico: Medico attivo, Medico licenziato

Un medico può essere attivo o licenziato quindi l'entità genitore è medico mentre le entità figlie sono Medico attivo e Medico licenziato. La generalizzazione è totale in quanto un medico può essere solo attivo o licenziato nella realtà di interesse.

3.3.4 Segretaria: Segretaria attiva, Segretaria licenziata

Una segretaria può essere attiva o licenziata quindi l'entità genitore è segretaria mentre le entità figlie sono Segretaria attiva e Segretaria licenziata. La generalizzazione è totale in quanto una segretaria può essere solo attiva o licenziata nella realtà di interesse.

3.4 Descrizione dello schema concettuale in forma grafica: lo schema E-R



3.5 Dizionario dei dati: Entità

Entità	Descrizione	Attributi	Identificatore
Persona	Medico, segretaria o paziente dello studio dentistico.	codice fiscale, nome, cognome, telefono, data_nascita, residenza(indirizzo, comune, provincia) sesso, e-mail	Codice fiscale
Personale	Personale dello studio dentistico	data_assunzione	
Medico	Medico che opera, o ha operato, nello studio.		
Medico Attivo	Medico che attualmente lavora nello studio		

Medico licenziato	Medico che ha lavorato nello studio in passato.	data_licenziamento	
Segretaria	Segretaria che lavora, o ha lavorato nello studio		
Segretaria attiva	Segretaria che attualmente lavora nello studio		
Segretaria licenziata	Segretaria che ha lavorato nello studio in passato	data_licenziamento	
Paziente	Paziente dello studio dentistico		
Account	Indica un account di una persona (paziente, medico, segretaria)	Username, password, privilegi	Username
Visita	Rappresenta una visita prenotata	Id_visita, data,ora,	id_visita
Sezione Visita	Rappresenta le sezioni in cui una visita è divisa		Chiave esterna da tipoVisita e id_visita
TipoVisita	Tipologia di visita	nome, prezzo, descrizione	nome

3.6 Dizionario dei dati: Relationship

Relazione	Descrizione	Entità coinvolte	Attributi
Possesso	Associa un paziente al proprio account.	Paziente (1,1) Account (1,1)	
Possesso	Associa una segretaria attiva al proprio account.	Segretaria attiva (1,1) Account (1,1)	
Possesso	Associa un medico attivo al proprio account.	Medico attivo (1,1) Account (1,1)	
Prenotazione	Associa un paziente alle visite che quest'ultimo ha prenotato.	Paziente (0,N) Visita (1,1)	
Specializzazione	Associa un medico alle tipologie di visite che esso può effettuare	Medico (0,N) TipoVisita (0,N)	
Divisione	Ogni visita è divisa in una o più sezioni, corrispondenti alle	Visita (1,N) SezioneVisita(1,1)	

	tipologie di visita		
Appartenenza	Associa le sezioni delle visite prenotate, alla relativa tipologia di visita	SezioneVisita (1,1) TipoVisita(0,N)	
Effettuazione	Associa un medico alle sezioni delle visite che esso effettua	Medico (0,N) SezioneVisita(1,N)	

3.7 Vincoli non esprimibili nell'E-R

1. Non posso prenotare appuntamenti in momenti precedenti a quello corrente.
2. Un medico non può effettuare due visite contemporaneamente e un paziente non può partecipare a due visite contemporaneamente.
3. Un medico non può essere anche segretaria, e una segretaria non può essere anche medico.
4. Un medico non può essere licenziato e al tempo stesso avere visite prenotate (future) a cui vi partecipa da medico.
5. Un medico che partecipa ad una visita deve essere specializzato in almeno una tipologia delle operazioni della visita alla quale partecipa come medico.

4) Progettazione logica

4.1 Analisi delle prestazioni su schemi E-R

4.1.1 tavola dei volumi ipotizzata dopo un anno di utilizzo di YouTooth

Concetto	Tipo	Volume
Persona	E	1000
Personale	E	10
Medico	E	10
Medico Attivo	E	5
Medico licenziato	E	5
Segretaria	E	2
Segretaria attiva	E	1
Segretaria licenziata	E	1
Paziente	E	1000
Account	E	1000
Visita	E	12 000
Sezione Visita	E	24 000
TipoVisita	E	10
Possesso	R	1000
Possesso	R	5
Possesso	R	1
Prenotazione	R	12 000
Specializzazione	R	70
Divisione	E	24 000
Appartenenza	E	24 000
Effettuazione	E	24 000

12 000 = numero medio di visite effettuate in un anno =

= 50 visite al giorno * 5 giorni * 4 settimane * 12 mesi

24 000 = numero medio di sezioni di visite effettuate in un anno =

= numero medio di visite in un anno * numero medio di sezioni di visita per una visita =

= 12 000 * 2

70 = numero di specializzazioni =

= numero di medici * numero medio di tipologie associate a un

medico = 10 * 7

4.1.2 Tavola degli accessi

4.1.2.1 Prenotazione di una nuova visita

Numero di accessi	Lettura/ Scrittura	Entità / Relazione
1	S	Visita
1	S	SezioneVisita
1	S	Effettuazione

4.1.2.2 Per un certo medico trova i pazienti che sono stati visitati o che saranno visitati da tale medico

Numero di accessi	Lettura/ Scrittura	Entità / Relazione
1	L	Medico
5 000	L	Effettuazione
2500	L	Visita
200	L	Pazienti

5 000 = numero medio di sezioni di visita effettuate da un medico =
= numero medio di sezioni di visita in un anno / medici =
= 24 000 / 5

2 500 = numero medio di sezioni di visita effettuate da un medico /
numero medio di sezioni di visita per una visita = 5 000 / 2

200 = numero di pazienti / medici
= 1000 / 5

4.2 Scelte sulla ristrutturazione dello schema E-R

4.2.1 Analisi delle ridondanze

- Importo di una visita:

Un eventuale attributo importo su “Visita”, per identificare il costo totale della visita, sarebbe ridondante in quanto tale dato e' ricavabile dalla somma dei prezzi delle singole tipologie di visita che compongono la visita stessa.

4.2.2 Eliminazione degli attributi composti

L'attributo composto Residenza dell'entità Persona viene sostituito con gli attributi componenti ossia indirizzo, comune, provincia.

4.2.3 Eliminazione delle generalizzazioni

4.2.3.1 Medico(genitore) – Medico Licenziato (figlia 1) – Medico attivo (figlia 2)

Data la contestualità degli accessi al padre e alle figlie, si è deciso di accorpare le entità figlie nell'entità genitore.

Come prima conseguenza i campi aggiuntivi dell'entità Medico Licenziato (cioè data licenziamento) vengono spostati nell'entità genitore. In caso di medico attivo, l'attributo “data licenziamento” conterrebbe valore null; se un medico venisse licenziato, allora la data di licenziamento conterrebbe la data del licenziamento.

Come seconda conseguenza la relazione di possesso tra Medico Attivo e account viene spostata all'entità genitore. In caso di medico attivo sarà presente la relazione possesso con account, mentre in caso di medico licenziato il campo che rappresenta la relazione sarà settato a null, ovvero un medico licenziato non avrà un account.

4.2.3.2 Segretaria(genitore) – Segretaria Licenziata (figlia 1) – Segretaria attiva (figlia 2)

Data la contestualità degli accessi al padre e alle figlie, si è deciso di accorpare le entità figlie nell'entità genitore.

Come prima conseguenza i campi aggiuntivi dell'entità Segretaria Licenziata (cioè data licenziamento) vengono spostati nell'entità genitore. In caso di segretaria attivo, l'attributo conterrebbe valore null, in caso una segretaria venisse licenziata la data licenziamento conterrebbe la data del licenziamento.

Come seconda conseguenza la relazione di possesso tra Segretaria attiva e account viene spostata all'entità genitore. In caso di segretaria attiva sarà presente la relazione possesso con account, mentre in caso di segretaria licenziata il campo che rappresenta la relazione sarà settato a null.

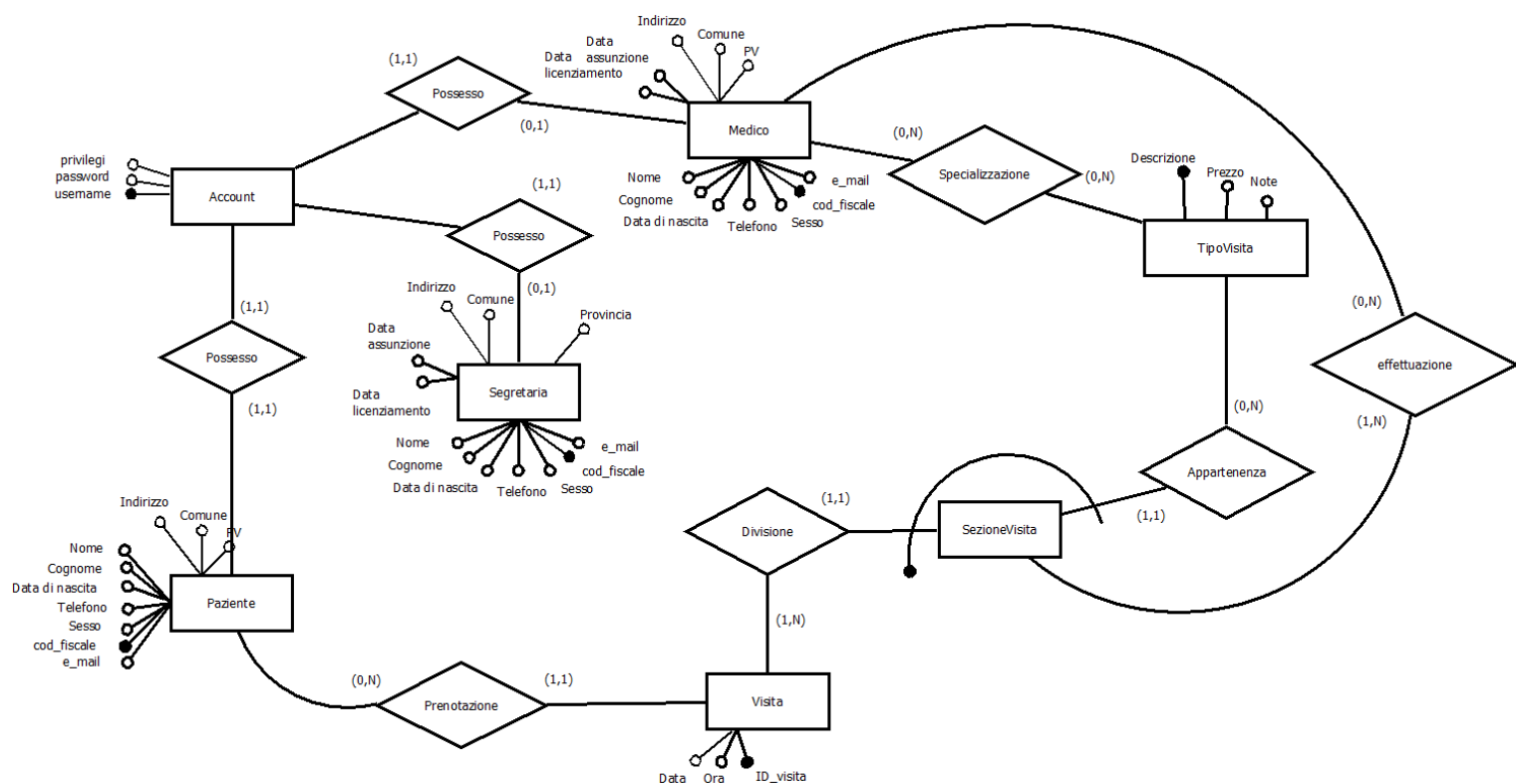
4.2.3.3 Persona (genitore) – Paziente (figlia) – Personale (figlia)

Dato che gli accessi alle figlie saranno effettuati in maniera distinta, si è deciso di accorpare l'entità genitore nelle due entità figlie. Come conseguenza gli attributi appartenenti all'entità Persona, in comune alle entità figlie, vengono ereditati dalle entità figlie.

4.2.3.4 Personale (genitore) – Medico (figlia) – Segretaria (figlia)

Dato che gli accessi alle figlie saranno effettuati in maniera distinta, si è deciso di accorpare l'entità genitore nelle due entità figlie. Come conseguenza gli attributi appartenenti all'entità Personale, vengono ereditati dalle entità figlie.

4.3 Schema E-R ristrutturato



4.4 Scelte sullo schema logico

A partire dallo schema E-R ristrutturato, siamo arrivati ad avere uno schema logico equivalente, in grado cioè di rappresentare le medesime informazioni dello schema E-R ristrutturato.

4.4.1 Entità e associazioni molti a molti:

- Per ogni entità, otteniamo una relazione con lo stesso nome, avente per attributi i medesimi attributi dell'entità e per chiave il suo identificatore.
- Per le associazioni molti a molti, otteniamo una relazione con lo stesso nome di tale associazione, come attributi, oltre ad eventuali attributi presenti specifici dell'associazione, gli identificatori delle entità coinvolte. Tali identificatori formano la chiave della relazione.

- Effettuazione (tra Medico e SezioneVisita, cardinalità N-N)

- Specializzazione (tra medico e TipoVisita, cardinalità N-N)

4.4.2 Associazioni uno a molti

- Paziente-Prenotazione-Visita:

E' stato scelto di fondere la relazione Visita e Prenotazione in un'unica relazione.

- Visita-Divisione-SezioneVisita:

E' stato scelto di fondere la relazione Divisione e SezioneVisita in un'unica relazione.

- SezioneVisita-Appartenenza-TipoVisita:

E' stato scelto di fondere la relazione Appartenenza con SezioneVisita in un'unica relazione.

4.4.3 Associazioni uno a uno

- Paziente-Possesso-Account

- Segretaria-Possesso-Account

- Medico-Possesso-Account

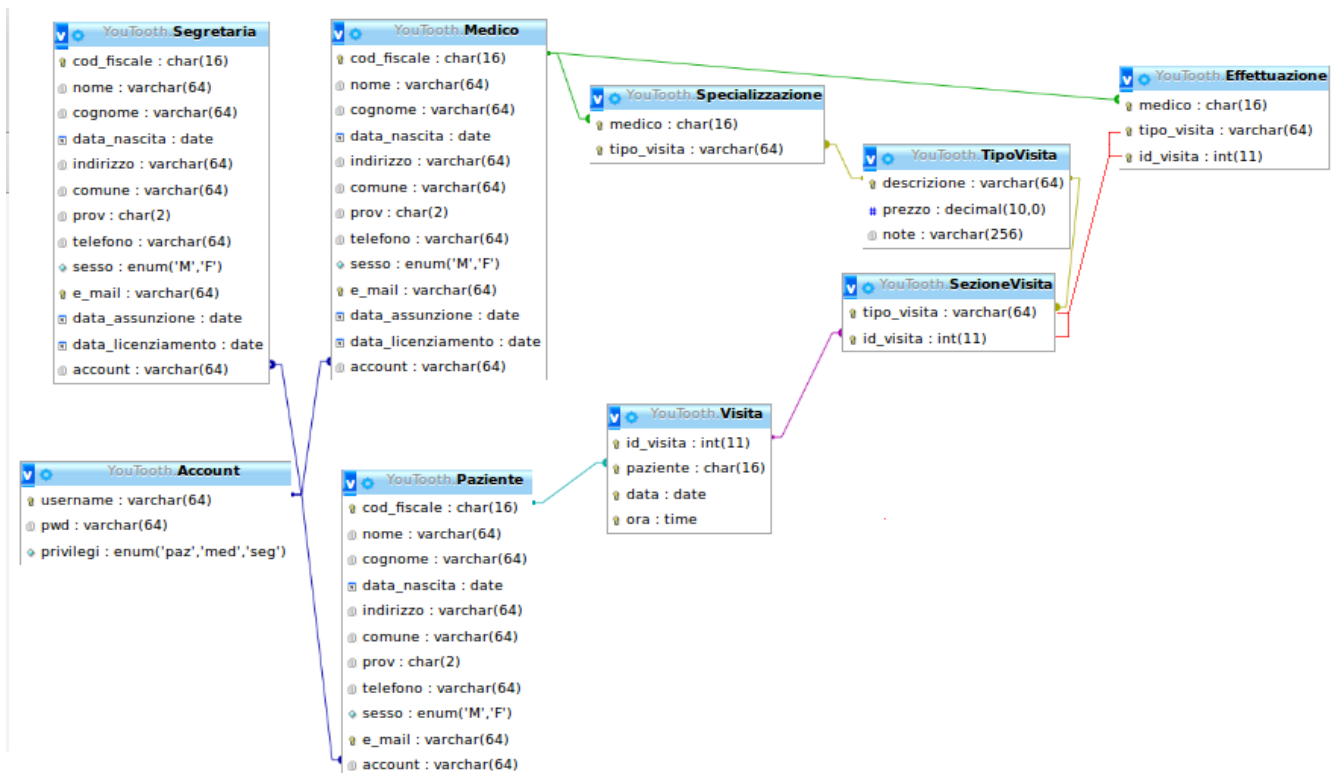
Per tutte e tre le associazioni, e' stato deciso di accorpare Paziente(/Segretaria/Medico) con Possesso.

4.5 Schema logico in forma testuale:

- Medico(cod_fiscale, nome, cognome, data_nascita, indirizzo, comune, prov, telefono, sesso, e_mail, data_assunzione, data_licenziamento, account)
 - PK(cod_fiscale)
 - account FK(Account)
- Segretaria(cod_fiscale, nome, cognome, data_nascita, indirizzo, comune, prov, telefono, sesso, e_mail, data_assunzione, data_licenziamento, account)
 - PK(cod_fiscale)
 - account FK(Account)
- Paziente(cod_fiscale, nome, cognome, data_nascita, indirizzo, comune, prov, telefono, sesso, e_mail, account)
 - PK(cod_fiscale)
 - account FK(Account)
- Account(username, pwd, privilegi)
 - PK(paziente)

- Visita(id_visita, paziente, data, ora)
 - PK(id_visita)
 - paziente FK(Paziente)
- SezioneVisita(tipo_visita, id_visita)
 - PK(tipo_visita, id_visita)
 - tipo_visita FK(TipoVisita)
 - id_visita FK(Visita)
- TipoVisita(descrizione, prezzo, note)
 - PK(descrizione)
- Specializzazione(medico, tipovisita)
 - PK(medico, tipo_visita)
 - medico FK(Medico)
 - tipo_visita FK(TipoVisita)
- Effettuazione(medico, tipo_visita, id_visita)
 - PK(medico, tipo_visita, id_visita)
 - medico FK(Medico)
 - tipo_visita, id_visita FK(SezioneVisita)

4.6 Schema logico in forma grafica



4.7 Vincoli semantici non catturati dallo schema logico

1. Non posso prenotare appuntamenti in momenti precedenti a quello corrente.
2. Devo avere, per ogni tipologia di visita relative a una certa visita, almeno un medico abilitato a tale tipologia che partecipa a quella visita.
3. Un medico non può effettuare due visite contemporaneamente e un paziente non può partecipare a due visite contemporaneamente.
4. Un medico non può essere anche segretaria, e una segretaria non può essere anche medico.
5. Un medico non può essere licenziato e al tempo stesso avere visite prenotate (future) a cui vi partecipa da medico.
6. Un medico che partecipa ad una visita deve essere specializzato in almeno una tipologia delle operazioni della visita alla quale partecipa come medico.

5) Definizione dello schema logico mediante DDL di MySql

```
/* Creazione della tabella Account */  
CREATE TABLE Account (  
    username VARCHAR(64) PRIMARY KEY,  
    pwd VARCHAR(64) NOT NULL,  
    privilegi ENUM('paz', 'med', 'seg') DEFAULT 'paz' NOT NULL  
);
```

```
/* Creazione della tabella Paziente */  
CREATE TABLE Paziente (  
    cod_fiscale CHAR(16) PRIMARY KEY,  
    nome VARCHAR(64) NOT NULL,  
    cognome VARCHAR(64) NOT NULL,  
    data_nascita DATE,  
    indirizzo VARCHAR(64),  
    comune VARCHAR(64),  
    prov CHAR(2),  
    telefono VARCHAR(64),  
    sesso ENUM('M', 'F'),  
    e_mail VARCHAR(64) UNIQUE,  
    account VARCHAR(64),  
    FOREIGN KEY (account)  
        REFERENCES Account(username)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

```

/* Creazione della tabella Medico */
CREATE TABLE Medico (
    cod_fiscale CHAR(16) PRIMARY KEY,
    nome VARCHAR(64) NOT NULL,
    cognome VARCHAR(64) NOT NULL,
    data_nascita DATE,
    indirizzo VARCHAR(64),
    comune VARCHAR(64),
    prov CHAR(2),
    telefono VARCHAR(64),
    sesso ENUM('M', 'F'),
    e_mail VARCHAR(64) UNIQUE,
    data_assunzione DATE,
    data_licenziamento DATE DEFAULT NULL,
    account VARCHAR(64),
    FOREIGN KEY (account)
        REFERENCES Account(username)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

```

```

/* Creazione della tabella Segretaria */
CREATE TABLE Segretaria (
    cod_fiscale CHAR(16) PRIMARY KEY,
    nome VARCHAR(64) NOT NULL,
    cognome VARCHAR(64) NOT NULL,
    data_nascita DATE,
    indirizzo VARCHAR(64),
    comune VARCHAR(64),
    prov CHAR(2),
    telefono VARCHAR(64),
    sesso ENUM('M', 'F'),
    e_mail VARCHAR(64) UNIQUE,
    data_assunzione DATE,
    data_licenziamento DATE DEFAULT NULL,
    account VARCHAR(64),
    FOREIGN KEY (account)
        REFERENCES Account(username)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

```

```

CREATE TABLE TipoVisita (
    descrizione VARCHAR(64) PRIMARY KEY,
    prezzo NUMERIC,
    note VARCHAR(256)
);

```

```

CREATE TABLE Specializzazione (
    medico CHAR(16),
    tipo_visita VARCHAR(64),
    PRIMARY KEY (medico, tipo_visita),
    FOREIGN KEY (medico) REFERENCES Medico (cod_fiscale) ON DELETE NO ACTION
ON UPDATE CASCADE,
    FOREIGN KEY (tipo_visita) REFERENCES TipoVisita (descrizione) ON DELETE NO
ACTION ON UPDATE CASCADE
);

```

```

CREATE TABLE Visita (
    id_visita INT AUTO_INCREMENT PRIMARY KEY,
    paziente CHAR(16),
    data DATE,
    ora TIME,
    UNIQUE (paziente, data, ora),
    FOREIGN KEY (paziente) REFERENCES Paziente (cod_fiscale) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE SezioneVisita (
    tipo_visita VARCHAR(64),
    id_visita INT,
    FOREIGN KEY (tipo_visita) REFERENCES TipoVisita (descrizione) ON DELETE NO
ACTION ON UPDATE CASCADE,
    FOREIGN KEY (id_visita) REFERENCES Visita (id_visita) ON DELETE CASCADE ON
UPDATE CASCADE,
    PRIMARY KEY (tipo_visita, id_visita)
);

```

```

CREATE TABLE Effettuazione (
    medico CHAR(16),
    tipo_visita VARCHAR(64),
    id_visita INT,
    FOREIGN KEY (tipo_visita, id_visita) REFERENCES SezioneVisita (tipo_visita, id_visita)
ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (medico) REFERENCES Medico (cod_fiscale) ON DELETE NO ACTION
ON UPDATE CASCADE,
    PRIMARY KEY (medico, tipo_visita, id_visita)
);

```


6) Query

6.1 Trovare le tipologie di visite che hanno avuto i due massimi numeri di richieste, mostrando anche il numero di volte in cui sono state richieste

/ QUERY 1 trovare le tipologie di visite che hanno avuto i due massimi numeri di richieste, mostrando anche il numero di volte in cui sono state richieste*/*

```
CREATE VIEW query1_aux AS
SELECT COUNT(*) AS cont, tv.descrizione
FROM TipoVisita as tv JOIN SezioneVisita as sv on tv.descrizione=sv.tipo_visita
GROUP BY tv.descrizione;
```

```
CREATE VIEW query1_aux2 AS
SELECT cont
FROM query1_aux
WHERE cont NOT IN
(
    SELECT MAX(query1_aux.cont)
    FROM query1_aux
);
```

```
CREATE VIEW query1 AS
SELECT tv.descrizione, count(*) as conteggio
FROM TipoVisita as tv JOIN SezioneVisita as sv on tv.descrizione=sv.tipo_visita
GROUP BY tv.descrizione
HAVING conteggio IN
(
    SELECT MAX(query1_aux.cont)
    FROM query1_aux
) OR conteggio IN
(
    SELECT MAX(query1_aux2.cont)
    FROM query1_aux2
);
```

6.2 Trovare i clienti che hanno fatto almeno una visita che contiene la tipologia di visita più costosa

/ QUERY 2 mostra i clienti che hanno fatto almeno una visita la cui tipologia è di prezzo più costoso*/*

```
CREATE VIEW query2 AS
SELECT DISTINCT p.cod_fiscale, p.cognome, p.nome
FROM Paziente as p
      JOIN Visita as v ON v.paziente=p.cod_fiscale
      JOIN SezioneVisita as sv ON sv.id_visita=v.id_visita
      JOIN TipoVisita as tv ON sv.tipo_visita=tv.descrizione
WHERE tv.descrizione IN
      (
        SELECT tv.descrizione
        FROM TipoVisita as tv
        WHERE tv.prezzo IN
              (
                SELECT MAX(tv.prezzo)
                FROM TipoVisita as tv
              )
      )
);
```

6.3 Trovare il cliente che ha fatto almeno una visita la cui tipologia e' di prezzo più costoso.

/ QUERY 3 il cliente che ha fatto almeno una visita, il cui prezzo è il secondo più costoso*/*

```
CREATE VIEW query3 AS
SELECT DISTINCT p.cod_fiscale, p.cognome, p.nome
FROM Paziente as p
    JOIN Visita as v ON v.paziente=p.cod_fiscale
    JOIN SezioneVisita as sv ON sv.id_visita=v.id_visita
    JOIN TipoVisita as tv ON sv.tipo_visita=tv.descrizione
WHERE tv.descrizione IN
(
    SELECT tv.descrizione
    FROM TipoVisita as tv
    WHERE tv.prezzo IN
    (
        SELECT MAX(tv.prezzo)
        FROM TipoVisita as tv
        WHERE tv.descrizione NOT IN
        (
            SELECT tv.descrizione
            FROM TipoVisita as tv
            WHERE tv.prezzo IN
            (
                SELECT MAX(tv.prezzo)
                FROM TipoVisita as tv
            )
        )
    )
);
```

6.4 Mostrare i medici e le segretarie che sono anche pazienti dello studio e che non si fanno visitare da più di 5 anni

*/*QUERY 4 mostrare i medici e le segretarie che sono anche pazienti dello studio e che non si fanno visitare da più di 5 anni */*

*/*ausiliaria: clienti che non si fanno piu vedere da 5 anni numero*/*

```
CREATE VIEW query4_aux_clienti_non_recenti AS
SELECT p.nome, p.cognome, p.cod_fiscale, max(v.data) as UltimaVisita
FROM Paziente as p JOIN Visita as v ON v.paziente=p.cod_fiscale
GROUP BY p.nome, p.cognome, p.cod_fiscale
HAVING UltimaVisita < DATE_SUB(CURRENT_DATE(), INTERVAL 4 YEAR);
```

```
CREATE VIEW query4 AS
SELECT DISTINCT m.cod_fiscale
FROM Medico AS m
WHERE m.cod_fiscale
IN (
    SELECT p.cod_fiscale
    FROM Paziente AS p
)
AND m.cod_fiscale
IN (
    SELECT query4_aux_clienti_non_recenti.cod_fiscale
    FROM query4_aux_clienti_non_recenti
)
UNION
SELECT DISTINCT s.cod_fiscale
FROM Segretaria AS s
WHERE s.cod_fiscale
IN (
    SELECT p.cod_fiscale
    FROM Paziente AS p
)
AND s.cod_fiscale
IN (
    SELECT query4_aux_clienti_non_recenti.cod_fiscale
    FROM query4_aux_clienti_non_recenti
);
```

6.5 Mostra i medici che hanno effettuato visite a pazienti che hanno fatto SOLO estrazioni di denti del giudizio.

*/*QUERY 5 mostra i medici che hanno effettuato visite a pazienti che hanno fatto SOLO estrazioni di denti del giudizio*/*

*/*ausiliaria: persone che hanno SOLO tolto denti del giudizio*/*

```
CREATE VIEW query5_aux_solo_giudiziosi AS
SELECT p.cod_fiscale
FROM Paziente as p
WHERE p.cod_fiscale NOT IN (
    SELECT p.cod_fiscale
    FROM Paziente as p JOIN Visita as v ON v.paziente=p.cod_fiscale
                                JOIN SezioneVisita as sv ON sv.id_visita=v.id_visita
    WHERE sv.tipo_visita <> 'Estrazione dente del giudizio'
);
```

```
CREATE VIEW query5 AS
SELECT DISTINCT m.cod_fiscale, m.cognome, m.nome
FROM Medico AS m
    JOIN Effettuazione as e on m.cod_fiscale=e.medico
    JOIN Visita as v on v.id_visita=e.id_visita
    JOIN Paziente as p on p.cod_fiscale=v.paziente
WHERE p.cod_fiscale IN
(
    SELECT query5_aux_solo_giudiziosi.cod_fiscale
    FROM query5_aux_solo_giudiziosi
);
```

6.6 Mostra la spesa totale di ogni cliente, mostrando cognome e nome del cliente, il codice fiscale e la sua spesa, ordinati per cognome e nome.

*/*QUERY 6 Mostra la spesa totale di ogni cliente, mostrando cognome e nome del cliente, il codice fiscale e la sua spesa, ordinati per cognome e nome.*/*

```
CREATE VIEW query6 AS
SELECT p.nome, p.cognome, p.cod_fiscale, totali(p.cod_fiscale)
FROM Paziente AS p
ORDER BY p.cognome, p.nome ASC;
```

7) View, Funzioni, Procedure, Triggers

7.1 Funzioni

7.1.1. is_operazione_disponibile

*/*La funzione restituisce TRUE se l'operazione "tipologia" e' disponibile il giorno "giorno", all'ora "orario". In caso contrario, restituisce FALSE.*/*

```
DELIMITER |
CREATE FUNCTION is_operazione_disponibile(tipologia VARCHAR(64), giorno DATE, orario
TIME)
RETURNS BOOLEAN
BEGIN
DECLARE cont INT(4);

SELECT COUNT(*) INTO cont
FROM Specializzazione as s
WHERE s.tipo_visita = tipologia AND s.medico NOT IN (
    SELECT e.medico
    FROM Effettuazione as e
        JOIN Visita as v ON (e.id_visita = v.id_visita)
    WHERE v.data = giorno AND v.ora = orario
    UNION
    SELECT m.cod_fiscale
    FROM Medico as m
    WHERE m.data_licenziamento IS NOT NULL
);

IF cont > 0 THEN
    RETURN TRUE;
ELSE
    RETURN FALSE;
END IF;

END
|
DELIMITER ;
```

7.1.2. prenotazioni_future

*/*La funzione restituisce il numero di visite prenotate con il medico medico, in un periodo successivo alla data di licenziamento di medico*/*

```
DELIMITER |
CREATE FUNCTION prenotazioni_future (medico CHAR(16), giorno_licenziamento DATE)
RETURNS INT
BEGIN
DECLARE conteggio INT;
SET conteggio = 0;
SELECT count(*) INTO conteggio
FROM Medico as m JOIN Effettuazione as e ON m.cod_fiscale=e.medico
JOIN SezioneVisita as sv ON (e.tipo_visita=sv.tipo_visita AND
e.id_visita=sv.id_visita)
JOIN Visita as v ON v.id_visita=sv.id_visita
WHERE v.data > giorno_licenziamento AND m.cod_fiscale=medico;
RETURN conteggio;
END
|
DELIMITER ;
```

7.1.3. totali

*/*FUNCTION dato in input un codice fiscale, trovare la somma totale di quanto hanno speso nelle visite */*

```
DELIMITER |
CREATE FUNCTION totali(cf VARCHAR(64))
RETURNS NUMERIC
BEGIN
DECLARE tot NUMERIC;

SELECT SUM(tv.prezzo) INTO tot
FROM Paziente as p
JOIN Visita as v ON v.paziente=p.cod_fiscale
JOIN SezioneVisita as sv ON sv.id_visita=v.id_visita
JOIN TipoVisita as tv ON tv.descrizione = sv.tipo_visita
WHERE p.cod_fiscale = cf;

RETURN tot;
END
|
DELIMITER ;
```

7.2 Procedure

7.2.1. ri_assumi

*/*La procedura ri_assumi effettua l'aggiunta di un nuovo account per un personale che ha già lavorato nello studio, oltre all'aggiornamento delle informazioni presenti su di esso nel DB.*/*

```
DELIMITER |
CREATE PROCEDURE ri_assumi(IN newcod_fiscale CHAR(16), newnome VARCHAR(64),
newcognome VARCHAR(64), newdata_nascita DATE, newindirizzo VARCHAR(64), newcomune
VARCHAR(64), newprov CHAR(2), newtelefono VARCHAR(64), newsesso CHAR(1), newe_mail
VARCHAR(64), newusername VARCHAR(64), newpassword VARCHAR(64), newprivilegi
CHAR(3))
BEGIN
    DECLARE conta INT(4);

    SELECT COUNT(*) INTO conta
    FROM Account
    WHERE newusername = username;

    IF conta = 0 THEN
        INSERT INTO Account (username, pwd, privilegi) VALUES
            (newusername, newpassword, newprivilegi);

        IF newprivilegi='med' THEN
            UPDATE Medico
            SET cod_fiscale=newcod_fiscale, nome=newnome, cognome=newcognome,
            data_nascita=newdata_nascita, indirizzo=newindirizzo, comune=newcomune, prov=newprov,
            telefono=newtelefono, sesso=newsesso, e_mail=newe_mail,
            data_assunzione=CURRENT_DATE(), data_licenziamento=NULL, account= newusername
            WHERE cod_fiscale=newcod_fiscale;
        ELSEIF newprivilegi='seg' THEN
            UPDATE Segretaria
            SET cod_fiscale=newcod_fiscale, nome=newnome, cognome=newcognome,
            data_nascita=newdata_nascita, indirizzo=newindirizzo, comune=newcomune, prov=newprov,
            telefono=newtelefono, sesso=newsesso, e_mail=newe_mail,
            data_assunzione=CURRENT_DATE(), data_licenziamento=NULL, account= newusername
            WHERE cod_fiscale=newcod_fiscale;
        END IF;
    END IF;
END
|
DELIMITER ;
```


7.2.2. assumi

*/*La procedura assumi effettua l'aggiunta di un nuovo account per un personale che non ha mai lavorato nello studio, oltre all'inserimento dei dati nella rispetta tabella "Segretaria" o "Medico", a seconda della tipologia che andiamo ad inserire.*/*

DELIMITER |

*CREATE PROCEDURE assumi(IN newcod_fiscale CHAR(16), newnome VARCHAR(64),
newcognome VARCHAR(64), newdata_nascita DATE, newindirizzo VARCHAR(64), newcomune
VARCHAR(64), newprov CHAR(2), newtelefono VARCHAR(64), newsesso CHAR(1), newe_mail
VARCHAR(64), newusername VARCHAR(64), newpassword VARCHAR(64), newprivilegi
CHAR(3))*

BEGIN

*INSERT INTO Account (username, pwd, privilegi) VALUES
(newusername, newpassword, newprivilegi);*

IF newprivilegi='med' THEN

*INSERT INTO Medico (cod_fiscale, nome, cognome, data_nascita, indirizzo,
comune, prov, telefono, sesso, e_mail, data_assunzione, data_licenziamento, account) VALUES*

*(newcod_fiscale,newnome,newcognome,newdata_nascita,newindirizzo,newcomune,newprov,newtel
efono,new Sesso, newe_mail, CURRENT_DATE(),NULL,newusername);*

ELSEIF newprivilegi='seg' THEN

*INSERT INTO Segretaria (cod_fiscale, nome, cognome, data_nascita,
indirizzo, comune, prov, telefono, sesso, e_mail, data_assunzione, data_licenziamento, account)
VALUES*

*(newcod_fiscale,newnome,newcognome,newdata_nascita,newindirizzo,newcomune,newprov,newtel
efono,new Sesso, newe_mail, CURRENT_DATE(),NULL,newusername);*

END IF;

END

|

DELIMITER ;

7.2.3. *disponibile*

```
/*La procedura disponibile invoca la funzione is_operazione_disponibile(tipologia, giorno, orario)
e restituisce in disp l'output di tale funzione*/
DELIMITER |
CREATE PROCEDURE disponibile(IN tipologia VARCHAR(64), giorno DATE, orario TIME, OUT
disp BOOL)
BEGIN
    SET disp = is_operazione_disponibile(tipologia, giorno, orario);
END
|
DELIMITER ;
```

7.3 Triggers

7.3.1. check_data_ora_insert

```
/*  
Controllo bonta' della data di prenotazione della visita:  
- no prenotazioni passate  
- no prenotazioni in momenti in cui lo studio e' chiuso  
*/  
DELIMITER |  
CREATE TRIGGER check_data_ora_insert AFTER INSERT ON Visita FOR EACH ROW  
IF (DAYOFWEEK(NEW.data)='6' OR DAYOFWEEK(NEW.data)='7' OR NEW.data <  
CURRENT_DATE() OR NEW.ora < '09:00:00' OR NEW.ora > '18:00:00')  
THEN DELETE FROM Visita  
WHERE(NEW.ora = Visita.ora AND NEW.data=Visita.data AND NEW.paziente =  
Visita.paziente AND  
NEW.id_visita=Visita.id_visita);  
END IF;  
|  
DELIMITER ;
```

7.3.2. check_data_ora_update

```
/*  
Controllo bonta' della data di prenotazione della visita:  
- no prenotazioni passate  
- no prenotazioni in momenti in cui lo studio e' chiuso  
*/  
DELIMITER |  
CREATE TRIGGER check_data_ora_update AFTER UPDATE ON Visita FOR EACH ROW  
IF (DAYOFWEEK(NEW.data)='6' OR DAYOFWEEK(NEW.data)='7' OR NEW.data <  
CURRENT_DATE() OR NEW.ora < '09:00:00' OR NEW.ora > '18:00:00')  
THEN DELETE FROM Visita  
WHERE(NEW.ora = Visita.ora AND NEW.data=Visita.data AND NEW.paziente =  
Visita.paziente AND  
NEW.id_visita=Visita.id_visita);  
END IF;  
|  
DELIMITER ;
```

7.3.3. segretaria_non_medico_insert

```
/*  
Una segretaria non può essere registrata anche come medico.  
*/  
DELIMITER |  
    CREATE TRIGGER segretaria_non_medico_insert BEFORE INSERT ON Segretaria FOR  
EACH ROW  
    IF (NEW.cod_fiscale IN (SELECT cod_fiscale FROM Medico WHERE  
cod_fiscale=NEW.cod_fiscale))  
    THEN /*DELETE FROM Medico WHERE(NEW.cod_fiscale = Medico.cod_fiscale)*/  
    INSERT INTO Medico  
        SELECT * FROM Medico LIMIT 1;  
    END IF;  
|  
DELIMITER ;
```

7.3.4. segretaria_non_medico_update

```
/*  
Una segretaria non può essere registrata anche come medico.  
*/  
DELIMITER |  
    CREATE TRIGGER segretaria_non_medico_update BEFORE UPDATE ON Segretaria  
FOR EACH ROW  
    IF (NEW.cod_fiscale IN (SELECT cod_fiscale FROM Medico WHERE  
cod_fiscale=NEW.cod_fiscale))  
    THEN INSERT INTO Medico  
        SELECT * FROM Medico LIMIT 1;  
    END IF;  
|  
DELIMITER ;
```

7.3.5. medico_non_segretaria_insert

```
/*  
Un medico non può essere registrato anche come segretaria.  
*/  
DELIMITER |  
    CREATE TRIGGER medico_non_segretaria_insert BEFORE INSERT ON Medico FOR  
EACH ROW  
    IF (NEW.cod_fiscale IN (SELECT cod_fiscale FROM Segretaria WHERE  
cod_fiscale=NEW.cod_fiscale))  
    THEN INSERT INTO Medico  
        SELECT * FROM Medico LIMIT 1;  
    END IF;  
|  
DELIMITER ;
```

7.3.6. medico_non_segretaria_update

```
/*  
Un medico non può essere registrato anche come segretaria.  
*/  
DELIMITER |  
CREATE TRIGGER medico_non_segretaria_update BEFORE UPDATE ON Medico FOR  
EACH ROW  
IF (NEW.cod_fiscale IN (SELECT cod_fiscale FROM Segretaria WHERE  
cod_fiscale=NEW.cod_fiscale))  
THEN INSERT INTO Medico  
SELECT * FROM Medico LIMIT 1;  
END IF;  
|  
DELIMITER ;
```

7.3.7. check_prezzo_positivo_ins

```
/*  
Una tipologia di visita non puo' avere un prezzo negativo.  
*/  
DELIMITER |  
CREATE TRIGGER check_prezzo_positivo_ins AFTER INSERT ON TipoVisita FOR EACH ROW  
IF NEW.prezzo < 0 THEN  
DELETE FROM TipoVisita  
WHERE NEW.Prezzo = TipoVisita.prezzo;  
END IF;  
|  
DELIMITER ;
```

7.3.8. check_prezzo_positivo_upd

```
/*  
Una tipologia di visita non puo' avere un prezzo negativo.  
*/  
DELIMITER |  
CREATE TRIGGER check_prezzo_positivo_ins AFTER INSERT ON TipoVisita FOR EACH ROW  
IF NEW.prezzo < 0 THEN  
DELETE FROM TipoVisita  
WHERE NEW.Prezzo = TipoVisita.prezzo;  
END IF;  
|  
DELIMITER ;
```

7.3.9. check_medico_licenziato_libero_insert

```
/*controllo che prima di licenziare un medico, questo non abbia visite future prenotate
*/
DELIMITER |
CREATE TRIGGER check_medico_licenziato_libero_insert AFTER INSERT ON `Medico`
FOR EACH ROW
IF (prenotazioni_future(NEW.cod_fiscale, NEW.data_licenziamento)<>0)
THEN
UPDATE Medico
SET Medico.data_licenziamento = NULL
WHERE Medico.cod_fiscale = NEW.cod_fiscale;
END IF;
|
DELIMITER ;
```

7.3.10. check_medico_licenziato_libero_update

```
/*controllo che prima di licenziare un medico, questo non abbia visite future prenotate
*/
DELIMITER |
CREATE TRIGGER check_medico_licenziato_libero_update AFTER UPDATE ON Medico
FOR EACH ROW
IF (prenotazioni_future(NEW.cod_fiscale, NEW.data_licenziamento)<>0)
THEN
UPDATE Medico
SET Medico.data_licenziamento = NULL
WHERE Medico.cod_fiscale = NEW.cod_fiscale;
END IF;
|
DELIMITER ;
```

7.3.11. check_medico_libero_specializzato_insert

*/*controllo che un medico non abbia due visite nello stesso momento e che sia specializzato in quella tipologia in inserimento*/*

DELIMITER |

CREATE TRIGGER check_medico_libero_specializzato_insert AFTER INSERT ON

Effettuazione

FOR EACH ROW

IF NEW.medico NOT IN (

SELECT Specializzazione.medico

FROM Specializzazione

WHERE Specializzazione.tipo_visita = NEW.tipo_visita

) OR NEW.medico IN (

SELECT e1.medico

FROM (Effettuazione AS e1 JOIN Visita AS v1 ON (e1.id_visita = v1.id_visita))

JOIN

(Effettuazione AS e2 JOIN Visita AS v2 ON (e2.id_visita = v2.id_visita))

ON e1.medico=e2.medico

WHERE v1.data=v2.data AND v1.ora=v2.ora AND v1.id_visita != v2.id_visita

)

THEN

DELETE FROM Effettuazione

WHERE NEW.medico=Effettuazione.medico AND

NEW.tipo_visita=Effettuazione.tipo_visita AND NEW.id_visita=Effettuazione.id_visita;

END IF;

|

DELIMITER ;

7.3.12. check_medico_libero_specializzato_update

*/*controllo che un medico non abbia due visite nello stesso momento e che sia specializzato in quella tipologia in modifica*/*

DELIMITER |

CREATE TRIGGER check_medico_libero_specializzato_update AFTER UPDATE ON Effettuazione

FOR EACH ROW

IF NEW.medico NOT IN (

SELECT Specializzazione.medico

FROM Specializzazione

WHERE Specializzazione.tipo_visita = NEW.tipo_visita

) OR NEW.medico IN (

SELECT e1.medico

FROM (Effettuazione AS e1 JOIN Visita AS v1 ON (e1.id_visita = v1.id_visita))

JOIN

(Effettuazione AS e2 JOIN Visita AS v2 ON (e2.id_visita = v2.id_visita))

ON e1.medico=e2.medico

WHERE v1.data=v2.data AND v1.ora=v2.ora AND v1.id_visita != v2.id_visita

)

THEN

DELETE FROM Effettuazione

WHERE NEW.medico=Effettuazione.medico AND

NEW.tipo_visita=Effettuazione.tipo_visita AND NEW.id_visita=Effettuazione.id_visita;

END IF;

|

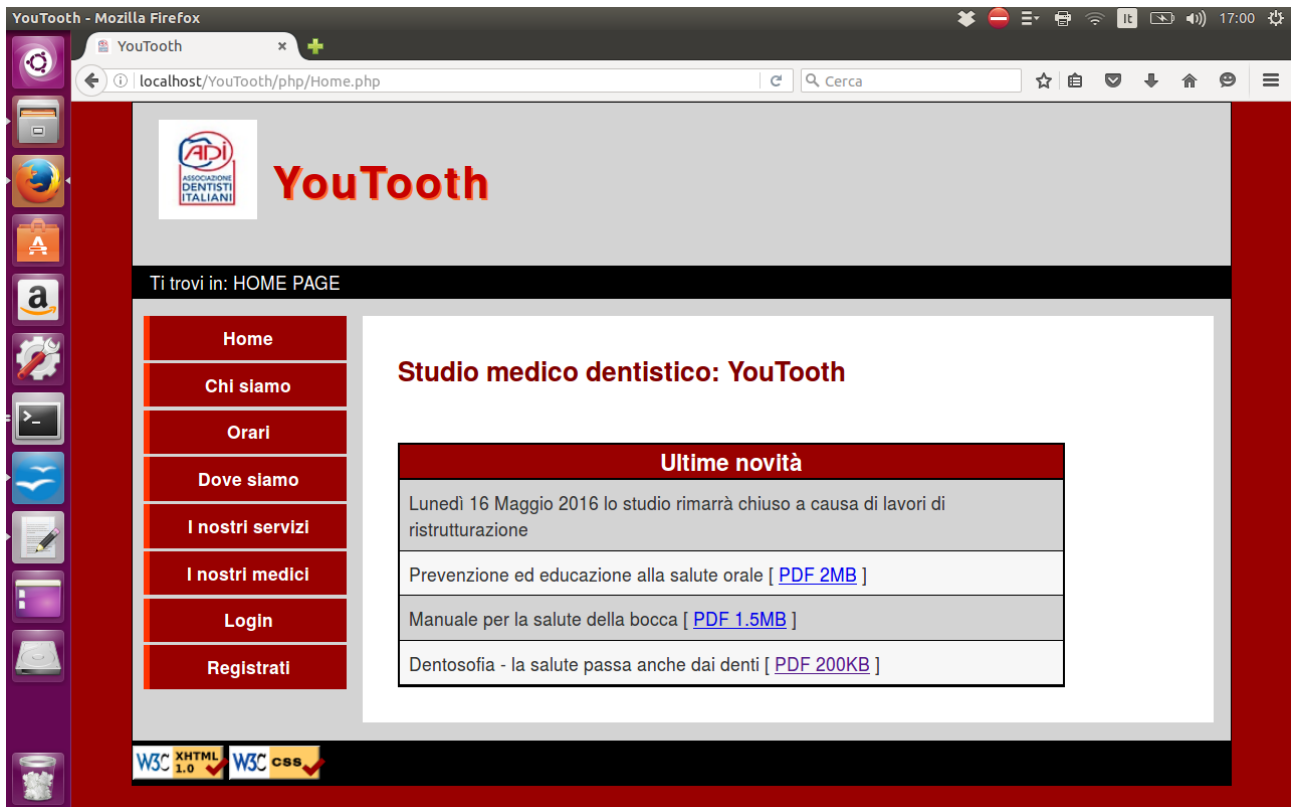
DELIMITER ;

8) Interfaccia WEB

8.1 Organizzazione delle pagine e operazioni:

GENERALE:

La Home Page del sito si presenta così:



- Questa e' la pagina vista da un utente esterno, che non ha effettuato il login.
- Per gli utenti loggati, cambieranno semplicemente alcune voci del menù laterale, con le operazioni che gli utenti possono eseguire sulla base dei loro privilegi.
- Descriviamo brevemente le varie operazioni disponibili:
 - HOME: e' la Home Page del sito, disponibile a tutti gli utenti, contiene le news relative allo studio.
 - Chi siamo: disponibile a tutti gli utenti, contiene una breve descrizione sulla storia dello studio dentistico.
 - Orari: disponibile a tutti gli utenti, contiene gli orari di apertura dello studio dentistico.
 - Dove siamo: disponibile a tutti gli utenti, contiene una mappa che mostra come raggiungere lo studio dentistico, oltre all'indirizzo dello studio odontoiatrico.
 - I nostri servizi, disponibile a tutti gli utenti, mostra i servizi attualmente offerti dallo studio dentistico.
 - I nostri medici, disponibile a tutti gli utenti, mostra i medici attualmente attivi nello studio e le loro relative specializzazioni.
 - Login: permette agli utenti già iscritti di effettuare l'accesso al sito.
 - Registrati: permette ai nuovi pazienti di registrarsi come tali allo studio dentistico.

I PAZIENTI:



The screenshot shows a web browser window with the URL `localhost/YouTooth/php/Mie_visite.php`. The page header includes the ADI logo and the site name 'YouTooth'. A navigation bar shows the current location: 'HOME PAGE > Le mie visite' and a welcome message: 'Benvenuto/a Daniel_paziente'.

On the left, there is a vertical menu with the following items: Home, Chi siamo, Orari, Dove siamo, I nostri servizi, I nostri medici, Il mio profilo, Le mie visite (selected), and Logout.

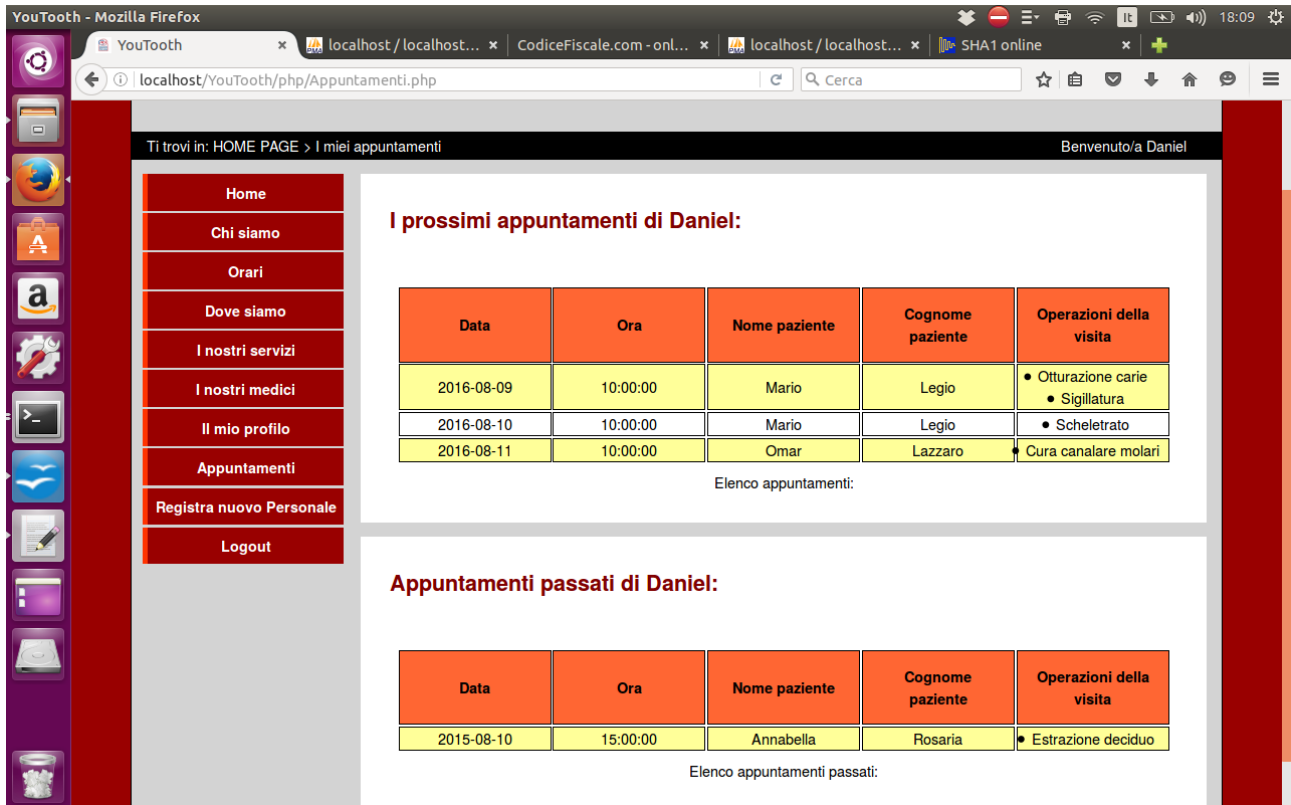
The main content area is titled 'Le visite di Daniel_paziente:' and contains a table of appointments:

Data	Ora	Elenco tipologia	Elenco medici	prezzo totale (€)
2016-08-08	10:00:00	<ul style="list-style-type: none">IgieneSbiancamento	<ul style="list-style-type: none">MariangelaAngela	270

Below the table, it says 'Elenco visite:'.

- Il mio profilo: Un paziente che ha effettuato l'accesso al sito, potrà inoltre visualizzare e modificare le proprie informazioni personali, fornite in fase di registrazione. Quest'operazione è presente per i pazienti dello studio, per i medici e per le segretarie.
- Le mie visite: Tramite questa operazione, il paziente potrà visualizzare le proprie visite prenotate, visualizzando data e orario della visita, le operazioni specifiche della visita, i medici che la effettuano, e il prezzo totale della visita.
- Logout: infine, come per ogni utente che ha effettuato il Login, è possibile disconnettersi dal sito, effettuando il logout.

I MEDICI:



Benvenuto/a Daniel

Ti trovi in: HOME PAGE > I miei appuntamenti

I prossimi appuntamenti di Daniel:

Data	Ora	Nome paziente	Cognome paziente	Operazioni della visita
2016-08-09	10:00:00	Mario	Legio	• Otturazione carie • Sigillatura
2016-08-10	10:00:00	Mario	Legio	• Scheletrato
2016-08-11	10:00:00	Omar	Lazzaro	• Cura canalare molari

Elenco appuntamenti:

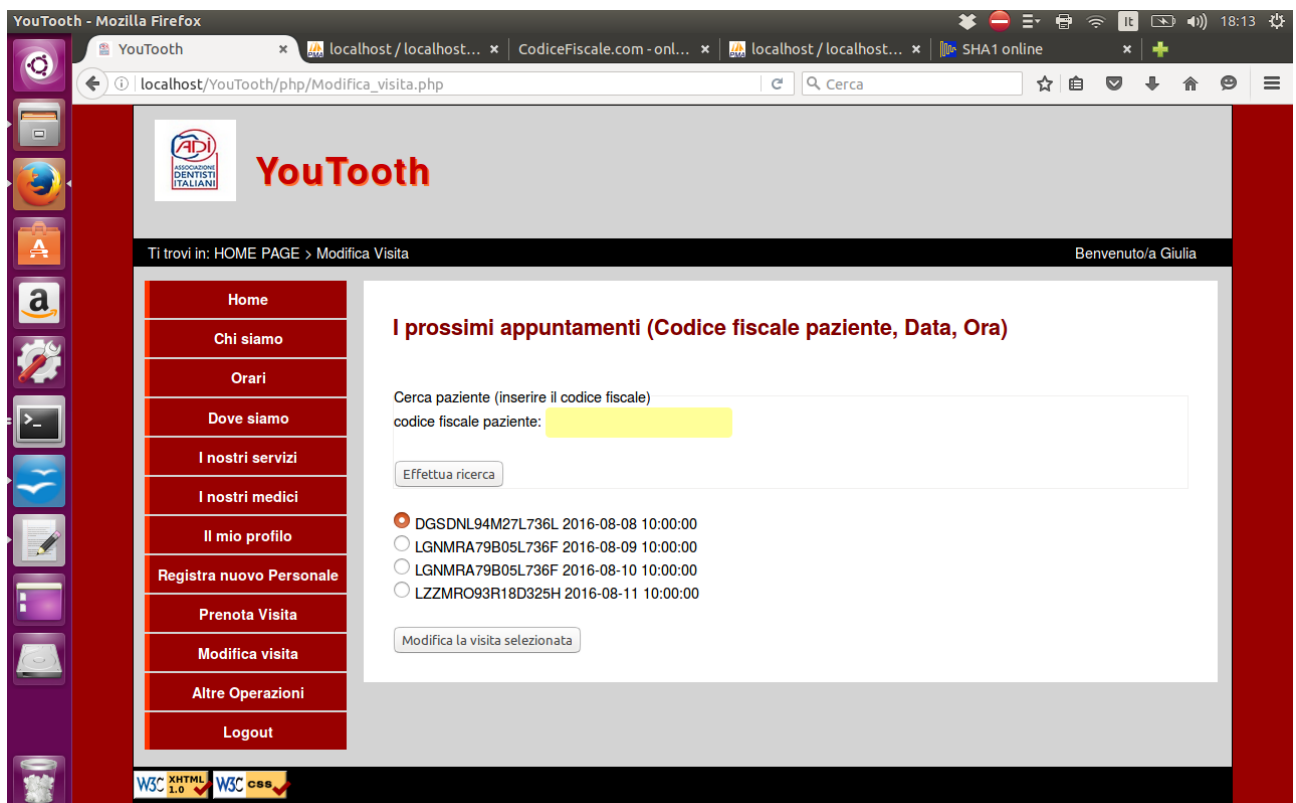
Appuntamenti passati di Daniel:

Data	Ora	Nome paziente	Cognome paziente	Operazioni della visita
2015-08-10	15:00:00	Annabella	Rosaria	• Estrazione deciduo

Elenco appuntamenti passati:

- Appuntamenti: Un medico che ha effettuato l'accesso potrà visualizzare tramite quest'operazione l'elenco degli appuntamenti futuri e passati ai quali partecipa.
- Registra nuovo personale: un medico può registrare nuovo personale per lo studio, sia segretarie, sia medici. Quest'operazione è presente anche per le segretarie.

LA SEGRETARIA:



Benvenuto/a Giulia

Ti trovi in: HOME PAGE > Modifica Visita

YouTooth

I prossimi appuntamenti (Codice fiscale paziente, Data, Ora)

Cerca paziente (inserire il codice fiscale)
codice fiscale paziente:

Effettua ricerca

☒ DGSDNL94M27L736L 2016-08-08 10:00:00
☐ LGNMRA79B05L736F 2016-08-09 10:00:00
☐ LGNMRA79B05L736F 2016-08-10 10:00:00
☐ LZZMRO93R18D325H 2016-08-11 10:00:00

Modifica la visita selezionata

- Prenota visita: tramite quest'operazione, la segretaria potrà prenotare nuove visite per i pazienti già registrati nello studio.
- Modifica visita: con quest'operazione, e' possibile modificare i dati relativi ad una prenotazione precedentemente effettuata.
- Altre Operazioni:
 - Aggiungi tipologia specializzazione: Permette di aggiungere all'elenco delle tipologie di visite nel DB un nuovo tipo visita.
 - Aggiungi specializzazione medico: Permette di aggiungere specializzazioni ai medici attivi dello studio dentistico.
 - Licenzia personale: Permette di licenziare personale che attualmente lavora nello studio dentistico.
 - QueryX: Visualizza alcune statistiche riguardanti lo studio medico.

8.2 Autenticazione:

L'autenticazione di un utente viene eseguita mediante utilizzo delle sessioni.

All'autenticazione di un utente, le variabili di sessione \$username e \$privilegi (med = medico, seg = segretaria, paz = paziente).

```
$_SESSION['username']=$username;
$_SESSION['privilegi']=$privilegi;
```

Non tutte le pagine sono accessibili a tutte le categorie di utenti, distinti da utenti esterni (che non hanno effettuato il login), e utenti loggati a loro volta suddivisi sulla base dei privilegi dell'account. Le pagine che richiedono determinati privilegi saranno quindi protette da una richiesta di Login, o non saranno comunque accessibili.