

Laboratorio
Corso di Basi di Dati e Web
A.A. 2023-24

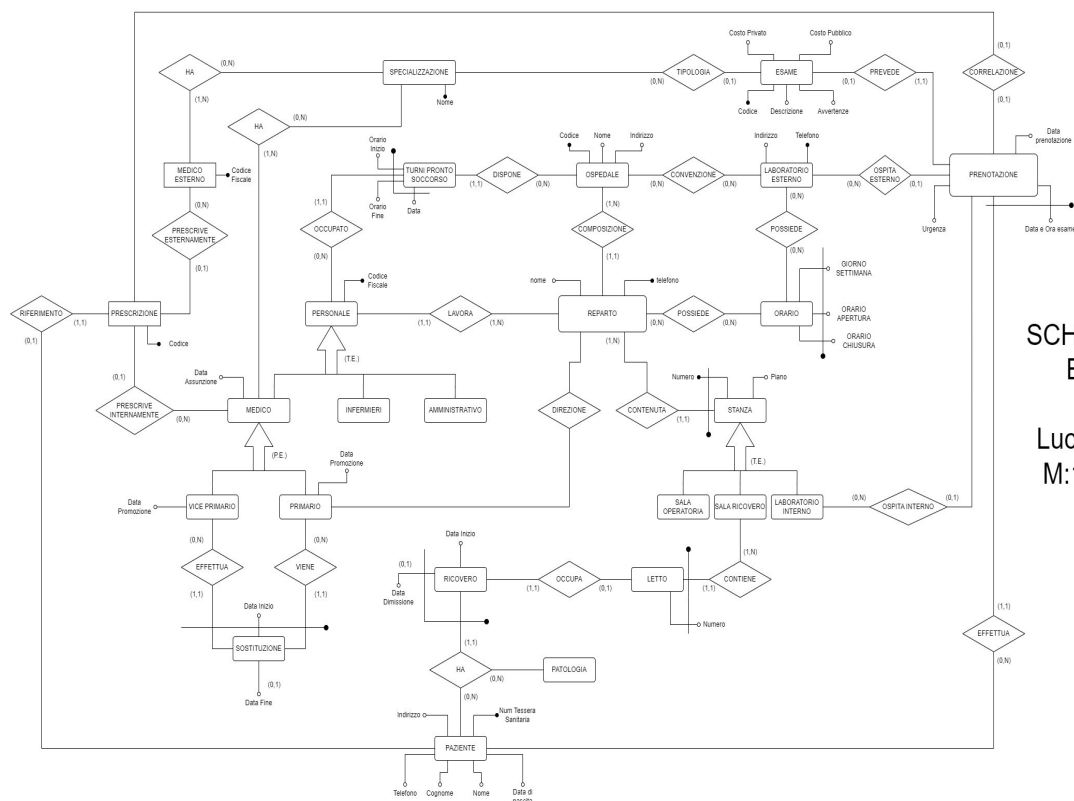
Progetto "Gestione Aziende Ospedaliere"

Autore:

Luca Alessi luca.alessi@studenti.unimi.it

1. Progettazione concettuale

1.1 Schema ER



SCHEMA ER
BASE

-
Luca Alessi
M:13286A

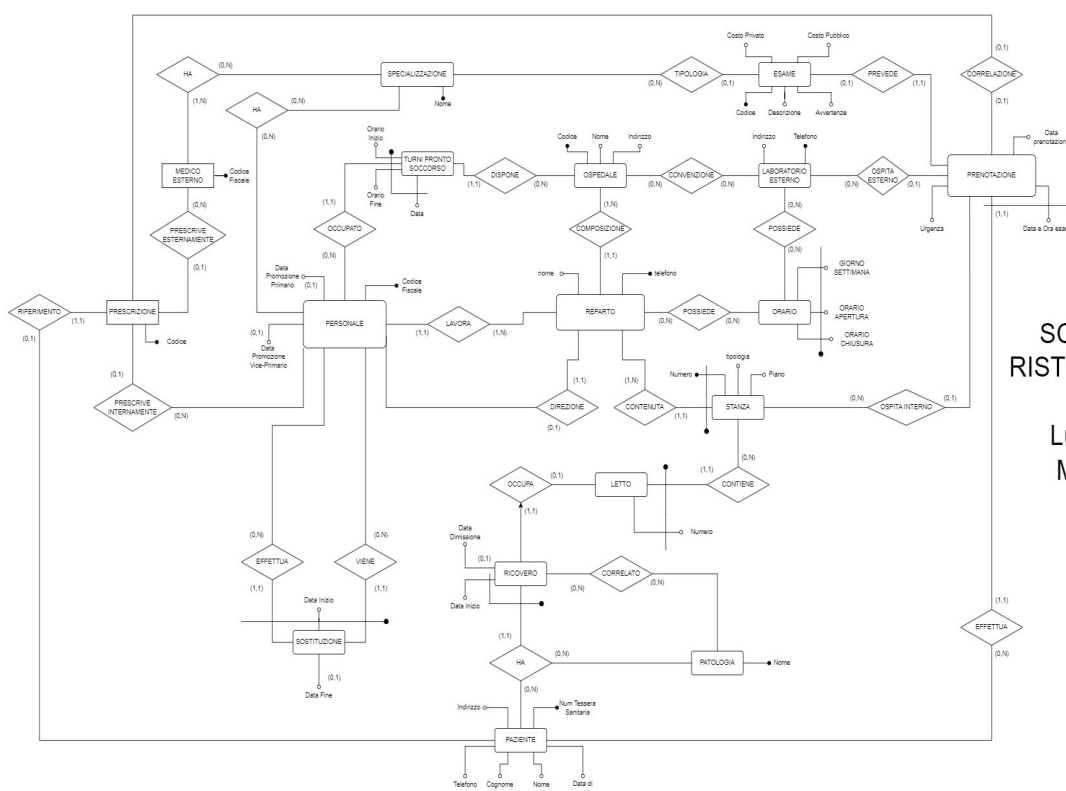
1.2 Vincoli di dominio

1. I turni del pronto soccorso devono assicurare continuità di servizio, non ci possono essere fasce orarie senza personale;
2. Il personale che opera in pronto soccorso deve essere medico o infermiere;
3. In ogni fascia oraria deve esserci almeno un medico;
4. Ogni indirizzo deve rispettare questa formattazione: ('nome via', 'nr civico', 'Citta');
5. I vari codici sono delle combinazioni di numeri e caratteri di lunghezza fissa;
6. Il vice primario per sostituire il primario deve lavorare nello stesso reparto;
7. La data di inizio sostituzione deve essere minore della data di fine;
8. Gli orari di chiusura devono essere maggiori degli orari di apertura;
9. La data di inizio del ricovero deve essere minore della data di fine;
10. Un paziente può avere un solo ricovero alla volta;
11. Una prescrizione deve essere prescritta da uno tra personale medico o medico esterno, non da entrambi;
12. Se un esame con specializzazione è stato prescritto, il medico che lo ha prescritto deve avere la stessa specializzazione dell'esame;
13. Le fasce di urgenza di una prenotazione sono: 'Verde', 'Gialla', 'Rossa';

14. L'esame relativo ad una prenotazione si può svolgere in uno tra i laboratori interni o quelli convenzionati, non in entrambi.

2. Progettazione logica

2.1 Schema ER ristrutturato



SCHEMA ER
RISTRUTTURATO

Luca Alessi
M:13286A

2.2 Vincoli di dominio

1. I ruoli del personale sono: 'Amministrazione', 'Infermiere', 'Medico';
2. Solo un medico può diventare vice o primario;
3. Solo un primario può dirigere un reparto;
4. Solo un primario può essere sostituito;
5. Solo un vice-primario può sostituire;
6. Un primario deve avere la data di promozione a viceprimario nulla;
7. Un viceprimario deve avere la data di promozione a primario nulla;
8. Solo i medici (quindi anche vice e primari) hanno specializzazioni;
9. Gli utilizzi di una stanza sono: 'Ambulatorio', 'Ricovero', 'Sala operatoria';
10. Se una stanza ha dei letti allora è una sala ricoveri;
11. Solo un ambulatorio può ospitare un esame;

2.3 Modello relazionale

- **OSPEDALE**(Codice, Nome, Indirizzo)
- **REPARTO**(Telefono, Nome, Ospedale, Direttore)
Ospedale REFERENCES OSPEDALE(Codice)
- **STANZA**(NumeroStanza, Reparto, Piano, Tipologia)
Reparto REFERENCES REPARTO(Telefono)
- **ORARIO**(Apertura, Chiusura, Giorno)
- **REPARTOPOSSIEDEORARIO**(Reparto, Apertura, Chiusura, Giorno)
Reparto REFERENCES REPARTO(Telefono)
Apertura REFERENCES ORARIO (Apertura)
Chiusura REFERENCES ORARIO (Chiusura)
Giorno REFERENCES ORARIO (Giorno)
- **AMBULATORIOESTERNO**(Telefono, Indirizzo)
- **ORARIOAMBULATORIOESTERNO**(Ambulatorio, Apertura, Chiusura, Giorno)
Ambulatorio REFERENCES AMBULATORIOESTERNO(Telefono)
Apertura REFERENCES ORARIO (Apertura)
Chiusura REFERENCES ORARIO (Chiusura)
Giorno REFERENCES ORARIO (Giorno)
- **SPECIALIZZAZIONE**(Nome)
- **ESAME**(Codice, Descrizione, CostoPubblico, CostoPrivato, Specializzazione)
Specializzazione REFERENCES SPECIALIZZAZIONE(Nome)
- **PATOLOGIA**(Nome)
- **PAZIENTE**(NumTesseraSanitaria, Indirizzo, Telefono, Nome, Cognome, DataNascita)
- **PAZIENTEHAPATOLOGIA**(NumTesseraSanitaria, Patologia)
NumTesseraSanitaria REFERENCES PAZIENTE(NumTesseraSanitaria)
Patologia REFERENCES PATOLOGIA(Nome)
- **LETTO**(Stanza, Reparto, NumeroLetto)
Stanza REFERENCES STANZA(Numero)
Reparto REFERENCES STANZA(Telefono)
- **PERSONALE**(CodiceFiscale, Reparto, Ruolo, DataPromozionePrimario, DataPromozioneVicePrimario, DataAssunzione)
Reparto REFERENCES REPARTO(Telefono)
- **MEDICOESTERNO**(CodiceFiscale)
- **MEDICOESTERNOHASPECIALIZZAZIONE**(MedicoEsterno, Specializzazione)
MedicoEsterno REFERENCES MEDICOESTERNO(CodiceFiscale)
Specializzazione REFERENCES SPECIALIZZAZIONE(Nome)
- **PRESCRIZIONE**(Codice, Paziente, MedicoInterno, MedicoEsterno)
Paziente REFERENCES PAZIENTE(NumTesseraSanitaria)
MedicoInterno REFERENCES PERSONALE(CodiceFiscale)
MedicoEsterno REFERENCES MEDICOESTERNO(CodiceFiscale)
- **PRENOTAZIONE**(Urgenza, DataPrenotazione, DataOraEsame, Paziente, AmbulatorioEsterno, NumeroStanza, Reparto, Esame, Prescrizione)
AmbulatorioEsterno REFERENCES AMBULATORIO ESTERNO(Telefono)
Paziente REFERENCES PAZIENTE(NumTesseraSanitaria)

- NumeroStanza REFERENCES STANZA(Numero)
- Reparto REFERENCES STANZA(Telefono)
- Esame REFERENCES ESAME(Codice)
- Prescrizione REFERENCES PRESCRIZIONE(Codice)
- **RICOVERO**(DataInizio, Paziente, DataDimissione, Letto, NumeroStanza, Reparto)
 - Paziente REFERENCES PAZIENTE(NumTesseraSanitaria)
 - Letto REFERENCES LETTO(Numero)
 - NumeroStanza REFERENCES STANZA(Numero)
 - Reparto REFERENCES STANZA(Telefono)
- **RICOVEROPATOLOGIA**(Ricovero, Patologia, Paziente)
 - Ricovero REFERENCES RICOVERO(DataInizio)
 - Patologia REFERENCES PATOLOGIA(Nome)
 - Paziente REFERENCES PAZIENTE(NumTesseraSanitaria)
- **TURNOPRONTOSOCCORSO**(Giorno, OrarioInizio, OrarioFine, Personale, Ospedale)
 - Personale REFERENCES PERSONALE(CodiceFiscale)
 - Ospedale REFERENCES OSPEDALE(Codice)
- **SOSTITUZIONE**(DataInizio, Primario, VicePrimario, DataFine)
 - Primario REFERENCES PERSONALE(CodiceFiscale)
 - Vice-Primario REFERENCES PERSONALE(CodiceFiscale)
- **PRIMARIOHASPECIALIZZAZIONE**(Primario, Specializzazione)
 - Primario REFERENCES PERSONALE(CodiceFiscale)
 - Specializzazione REFERENCES SPECIALIZZAZIONE(Nome)
- **AVVERTENZE**(Descrizione)
- **ESAMEHAAVVERTENZA**(Esame, Avvertenza)
 - Esame REFERENCES ESAME(Codice)
 - Avvertenza REFERENCES AVVERTENZA(Descrizione)
- **CONVENZIONEOSPEDALEAMBULATORIO**(Ospedale, AmbulatorioEsterno)
 - Ospedale REFERENCES OSPEDALE(Codice)
 - AmbulatorioEsterno REFERENCES AMBULATORIOESTERNO(Telefono)

2.4 Codice SQL

```
CREATE TABLE OSPEDALE(  
    Codice CHAR(20) PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Indirizzo VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Reparto (  
    Telefono CHAR(20) PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Ospedale CHAR(20) NOT NULL,  
    Direttore CHAR(16),  
    UNIQUE(Nome, Ospedale),  
    FOREIGN KEY (Ospedale) REFERENCES Ospedale(Codice)  
);
```

```
CREATE TABLE STANZA (  
    NumeroStanza INTEGER CHECK (NumeroStanza > 0),  
    Reparto CHAR(20),  
    Piano INTEGER NOT NULL CHECK (Piano > 0),  
    Tipologia VARCHAR(255),  
    CHECK(Tipologia = 'Ambulatorio' OR Tipologia = 'Sala operatoria' OR Tipologia =  
    'Ricovero'),  
    PRIMARY KEY (NumeroStanza, Reparto),  
    FOREIGN KEY (Reparto) REFERENCES Reparto(Telefono)  
);
```

```
CREATE TABLE ORARIO(  
    Apertura TIME,  
    Chiusura TIME,  
    Giorno VARCHAR(9),  
    CHECK(Apertura < Chiusura),  
    PRIMARY KEY (Apertura, Chiusura, Giorno)  
);
```

```
CREATE TABLE REPARTOPOSSIEDEORARIO(  
    Reparto CHAR(20),  
    Apertura TIME,  
    Chiusura TIME,  
    Giorno VARCHAR(9),  
    CHECK (Apertura < Chiusura),  
    PRIMARY KEY (Reparto, Apertura, Chiusura, Giorno),  
    FOREIGN KEY (Reparto) REFERENCES REPARTO(Telefono),
```

```
FOREIGN KEY (Apertura, Chiusura, Giorno) REFERENCES ORARIO(Apertura, Chiusura,
Giorno)
);
```

```
CREATE TABLE AMBULATORIOESTERNO(
Telefono CHAR(13) PRIMARY KEY,
Indirizzo VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE ORARIOAMBULATORIOESTERNO(
Ambulatorio CHAR(9),
Apertura TIME,
Chiusura TIME,
Giorno VARCHAR(9),
CHECK (Apertura < Chiusura),
PRIMARY KEY (Ambulatorio, Apertura, Chiusura, Giorno),
FOREIGN KEY (Ambulatorio) REFERENCES AMBULATORIOESTERNO(Telefono),
FOREIGN KEY (Apertura, Chiusura, Giorno) REFERENCES ORARIO(Apertura, Chiusura,
Giorno)
);
```

```
CREATE TABLE SPECIALIZZAZIONE(
Nome VARCHAR(255) PRIMARY KEY
);
```

```
CREATE TABLE ESAME(
Codice CHAR(9) PRIMARY KEY,
Descrizione VARCHAR(255),
CostoPubblico DECIMAL(10,2) NOT NULL CHECK (CostoPubblico > 0),
CostoPrivato DECIMAL(10,2) NOT NULL CHECK (CostoPrivato > 0),
Specializzazione VARCHAR(255),
FOREIGN KEY (Specializzazione) REFERENCES SPECIALIZZAZIONE(Nome)
);
```

```
CREATE TABLE PATOLOGIA(
Nome VARCHAR(255) PRIMARY KEY
);
```

```
CREATE TABLE PAZIENTE(
NumTesseraSanitaria CHAR(20) PRIMARY KEY,
Indirizzo VARCHAR(255) NOT NULL,
Telefono CHAR(13) UNIQUE NOT NULL,
Nome VARCHAR(255) NOT NULL,
Cognome VARCHAR(255) NOT NULL,
```

```
DataNascita DATE NOT NULL,  
CHECK (LENGTH(NumTesseraSanitaria) = 20),  
CHECK (DataNascita <= CURRENT_DATE)  
);
```

```
CREATE TABLE PAZIENTEHAPATOLOGIA(  
    NumTesseraSanitaria CHAR(20),  
    Patologia VARCHAR(255),  
    PRIMARY KEY (NumTesseraSanitaria, Patologia),  
    FOREIGN KEY (NumTesseraSanitaria) REFERENCES PAZIENTE(NumTesseraSanitaria),  
    FOREIGN KEY (Patologia) REFERENCES PATOLOGIA(Nome)  
);
```

```
CREATE TABLE LETTO(  
    Stanza INTEGER,  
    Reparto CHAR(20),  
    NumeroLetto INTEGER CHECK (NumeroLetto > 0),  
    PRIMARY KEY (Stanza, Reparto, NumeroLetto),  
    FOREIGN KEY (Stanza, Reparto) REFERENCES STANZA(NumeroStanza, Reparto)  
);
```

```
CREATE TABLE PERSONALE(  
    CodiceFiscale CHAR(16) PRIMARY KEY,  
    Reparto CHAR(20),  
    Ruolo VARCHAR(255) NOT NULL,  
    DataPromozionePrimario DATE,  
    DataPromozioneVicePrimario DATE,  
    DataAssunzione DATE NOT NULL,  
    CHECK (Ruolo = 'Medico' OR Ruolo = 'Infermiere' OR Ruolo = 'Amministrativo'),  
    CHECK (DataAssunzione <= DataPromozionePrimario OR DataPromozionePrimario IS  
    NULL),  
    CHECK(DataAssunzione<= DataPromozioneVicePrimario OR DataPromozioneVicePrimario  
    IS NULL),  
    CHECK(DataPromozioneVicePrimario<=DataPromozionePrimario OR  
    DataPromozioneVicePrimario IS NULL),  
    FOREIGN KEY (Reparto) REFERENCES Reparto(Telefono)  
  
);
```

```
CREATE TABLE MEDICOESTERNO(  
    CodiceFiscale CHAR(16) PRIMARY KEY  
);
```



```

CREATE TABLE MEDICOESTERNOHASPECIALIZZAZIONE(
    MedicoEsterno CHAR(16),
    Specializzazione VARCHAR(255),
    PRIMARY KEY (MedicoEsterno, Specializzazione),
    FOREIGN KEY (MedicoEsterno) REFERENCES MEDICOESTERNO(CodiceFiscale),
    FOREIGN KEY (Specializzazione) REFERENCES SPECIALIZZAZIONE(Nome)
);

```

```

CREATE TABLE PRESCRIZIONE(
    Codice CHAR(20) PRIMARY KEY,
    Paziente CHAR(20),
    MedicoInterno CHAR(16),
    MedicoEsterno CHAR(16),
    CHECK (MedicoEsterno IS NULL AND MedicoInterno IS NOT NULL
    OR
    (MedicoEsterno IS NOT NULL AND MedicoInterno IS NULL)),
    FOREIGN KEY (MedicoInterno) REFERENCES PERSONALE(CodiceFiscale),
    FOREIGN KEY (MedicoEsterno) REFERENCES MEDICOESTERNO(CodiceFiscale),
    FOREIGN KEY (Paziente) REFERENCES PAZIENTE(NumTesseraSanitaria)
);

```

```

CREATE TABLE PRENOTAZIONE(
    Urgenza VARCHAR(6) NOT NULL,
    DataPrenotazione DATE NOT NULL,
    DataOraEsame TIMESTAMP,
    Paziente CHAR(20),
    AmbulatorioEsterno CHAR(9),
    NumeroStanza INTEGER,
    Reparto CHAR(20),
    Esame CHAR(20),
    Prescrizione CHAR(20) UNIQUE,
    CHECK (Urgenza = 'Verde' OR Urgenza = 'Giallo' OR Urgenza = 'Rosso'),
    CHECK (DataPrenotazione <= CURRENT_DATE),
    CHECK (DataPrenotazione <= DataOraEsame),
    CHECK (
    (NumeroStanza IS NULL AND Reparto IS NULL AND AmbulatorioEsterno IS NOT NULL)
    OR
    (NumeroStanza IS NOT NULL AND Reparto IS NOT NULL AND AmbulatorioEsterno IS
    NULL)
    ),
    UNIQUE(Esame, DataPrenotazione, Paziente),
    PRIMARY KEY (DataOraEsame, Paziente),
    FOREIGN KEY (AmbulatorioEsterno) REFERENCES AMBULATORIOESTERNO(Telefono),
    FOREIGN KEY (Paziente) REFERENCES PAZIENTE(NumTesseraSanitaria),

```

```
FOREIGN KEY (NumeroStanza, Reparto) REFERENCES STANZA(NumeroStanza, Reparto),  
FOREIGN KEY (Esame) REFERENCES ESAME(Codice),  
FOREIGN KEY (Prescrizione) REFERENCES PRESCRIZIONE(Codice)  
);
```

```
CREATE TABLE RICOVERO(  
    DataInizio DATE,  
    DataDimissione DATE,  
    Paziente CHAR(20),  
    Letto INTEGER,  
    NumeroStanza INTEGER,  
    Reparto CHAR(20),  
    CHECK (DataInizio <= DataDimissione OR DataDimissione IS NULL),  
    CHECK (DataInizio <= CURRENT_DATE),  
    CHECK (DataDimissione <= CURRENT_DATE OR DataDimissione IS NULL),  
    UNIQUE(Letto, NumeroStanza, Reparto),  
    PRIMARY KEY (DataInizio, Paziente),  
    FOREIGN KEY (Paziente) REFERENCES PAZIENTE(NumTesseraSanitaria),  
    FOREIGN KEY (NumeroStanza, Reparto, Letto) REFERENCES LETTO(Stanza, Reparto, NumeroLetto),  
    FOREIGN KEY (NumeroStanza, Reparto) REFERENCES STANZA(NumeroStanza, Reparto)  
);
```

```
CREATE TABLE RICOVEROPATOLOGIA(  
    Ricovero DATE,  
    Patologia VARCHAR(255),  
    Paziente CHAR(20),  
    PRIMARY KEY (Ricovero, Patologia, Paziente),  
    FOREIGN KEY (Ricovero, Paziente) REFERENCES RICOVERO(DataInizio, Paziente),  
    FOREIGN KEY (Patologia) REFERENCES PATOLOGIA(Nome),  
    FOREIGN KEY (Paziente) REFERENCES PAZIENTE(NumTesseraSanitaria)  
);
```

```
CREATE TABLE TURNOPRONTOSOCCORSO(  
    Giorno VARCHAR(9),  
    OrarioInizio TIME,  
    OrarioFine TIME,  
    Personale CHAR(16),  
    Ospedale CHAR(20) NOT NULL,  
    CHECK (OrarioInizio < OrarioFine),  
    PRIMARY KEY (Giorno, OrarioInizio, OrarioFine, Personale),  
    FOREIGN KEY (Ospedale) REFERENCES OSPEDALE(Codice),  
    FOREIGN KEY (Personale) REFERENCES PERSONALE(CodiceFiscale)  
);
```

```

CREATE TABLE SOSTITUZIONE(
    DataInizio DATE,
    Primario CHAR(16),
    VicePrimario CHAR(16),
    DataFine DATE,
    CHECK (DataInizio <= DataFine),
    CHECK (DataInizio <= CURRENT_DATE),
    CHECK (DataFine <= CURRENT_DATE),
    CHECK (Primario != VicePrimario),
    PRIMARY KEY (DataInizio, Primario, VicePrimario),
    FOREIGN KEY (Primario) REFERENCES PERSONALE(CodiceFiscale),
    FOREIGN KEY (VicePrimario) REFERENCES PERSONALE(CodiceFiscale)
);

```

```

CREATE TABLE PRIMARIOHASPECIALIZZAZIONE(
    Primario CHAR(16),
    Specializzazione VARCHAR(255),
    PRIMARY KEY (Primario, Specializzazione),
    FOREIGN KEY (Primario) REFERENCES PERSONALE(CodiceFiscale),
    FOREIGN KEY (Specializzazione) REFERENCES SPECIALIZZAZIONE(Nome)
);

```

```

CREATE TABLE AVVERTENZE(
    Descrizione VARCHAR(255) PRIMARY KEY
);

```

```

CREATE TABLE ESAMEHAAVVERTENZA(
    Esame CHAR(9),
    Avvertenza VARCHAR(255),
    PRIMARY KEY (Esame, Avvertenza),
    FOREIGN KEY (Esame) REFERENCES ESAME(Codice),
    FOREIGN KEY (Avvertenza) REFERENCES AVVERTENZE(Descrizione)
);

```

```

CREATE TABLE CONVENZIONEOSPEDALEAMBULATORIO(
    Ospedale CHAR(20),
    AmbulatorioEsterno CHAR(9),
    PRIMARY KEY (Ospedale, AmbulatorioEsterno),
    FOREIGN KEY (Ospedale) REFERENCES OSPEDALE(Codice),
    FOREIGN KEY (AmbulatorioEsterno) REFERENCES AMBULATORIOESTERNO(Telefono)
);

```

```

CREATE TABLE UTENTE(

```

```
username VARCHAR(255) PRIMARY KEY,  
password VARCHAR(255) NOT NULL,  
ruolo VARCHAR(255) NOT NULL,  
CHECK (ruolo = 'Medico Interno' OR ruolo = 'Medico Esterno' OR ruolo = 'Infermiere' OR  
ruolo = 'Paziente' OR ruolo = 'Amministrativo'),  
UNIQUE(username, password)  
);
```

Query SQL

1) Determinare i vice primari che hanno sostituito esattamente una volta il proprio primario

```
SELECT Viceprimario  
FROM Sostituzione  
GROUP BY Viceprimario  
HAVING COUNT(Viceprimario) = 1;
```

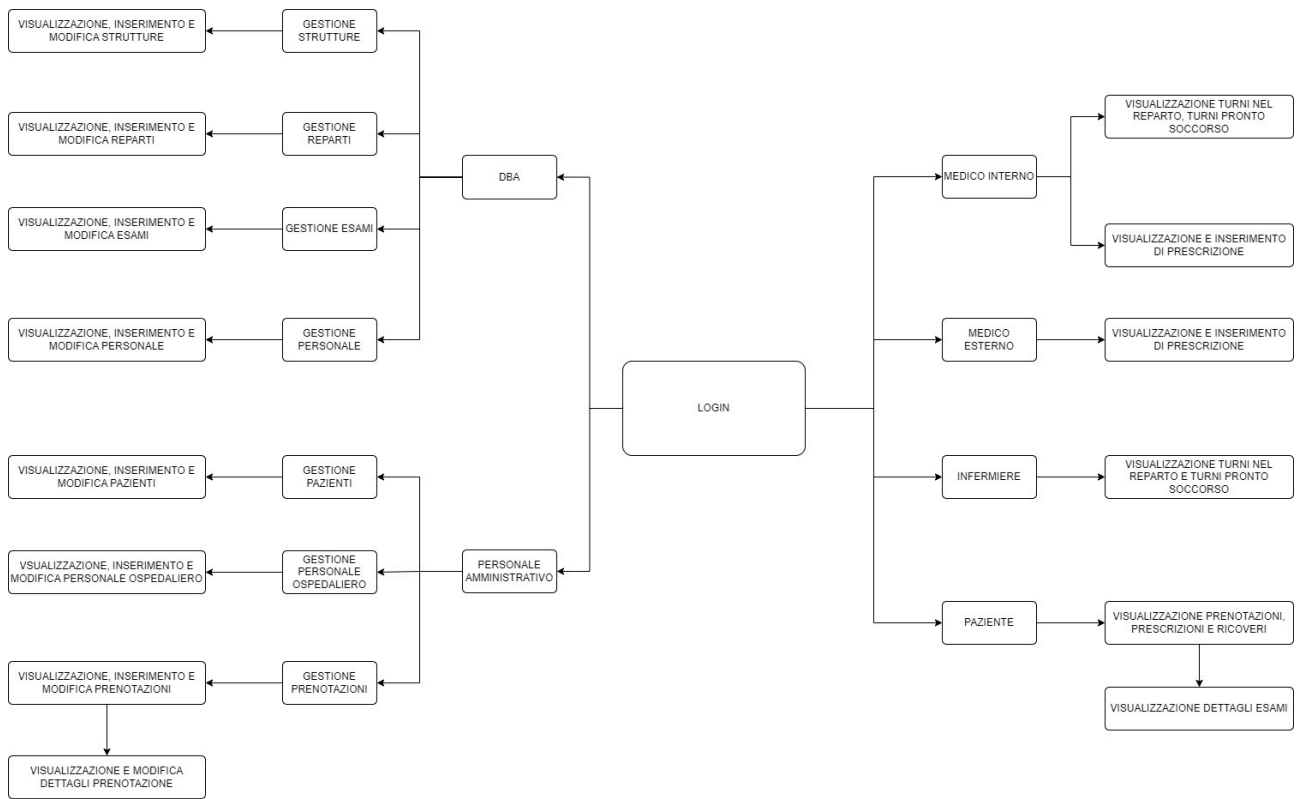
2) Determinare i vice primari che hanno sostituito almeno due volte il proprio primario

```
SELECT Viceprimario  
FROM Sostituzione  
GROUP BY Viceprimario  
HAVING COUNT(Viceprimario) >= 2;
```

3) Determinare i vice primari che non hanno mai sostituito il proprio primario

```
SELECT CodiceFiscale  
FROM Personale  
WHERE DataPromozioneViceprimario IS NOT NULL  
EXCEPT  
SELECT Viceprimario  
FROM Sostituzione;
```

3. Progettazione del sito



Nome del database PostgreSQL: progetto

Nome della cartella sotto htdocs: lab/progetto