

Laboratorio
Corso di Basi di Dati e Web
A.A. {2021/2022}

Progetto “{Gestione aziende ospedaliere}”

Autore:

{Diego} {Turri} {diego.turri@studenti.unimi.it}

Data di consegna: {19/07/2022}

Il progetto è già stato consegnato in precedenza? {no}

Se sì, indica tutte le date degli appelli in cui il progetto è stato consegnato:

**Se sì, indica brevemente (tre righe al massimo) le modifiche apportate rispetto
all'ultimo progetto consegnato:**

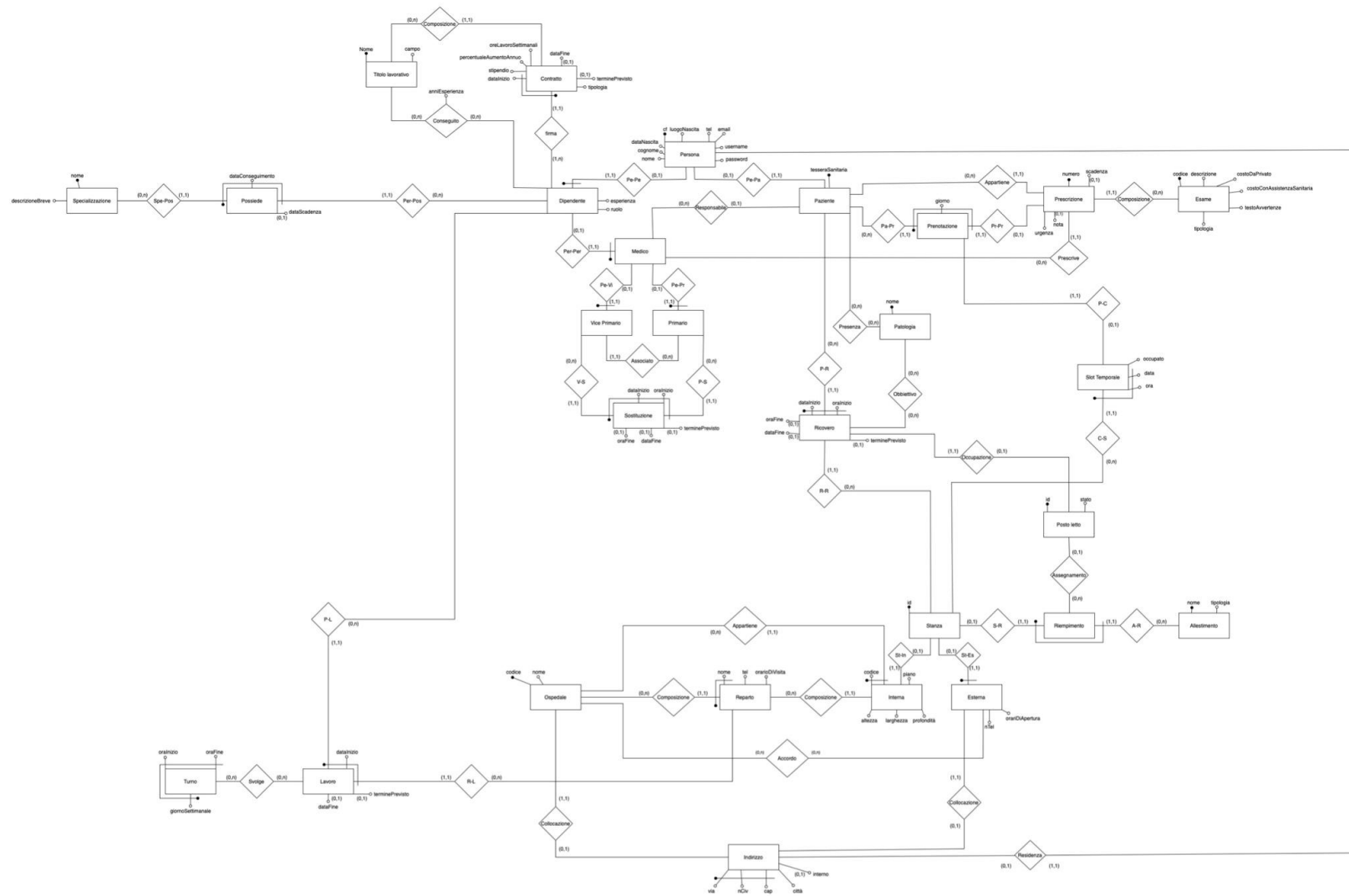
In questa sezione vanno riportate le conclusioni della progettazione concettuale. Non siamo interessati ai vari passaggi intermedi con i quali arrivate a costruire lo schema concettuale. Vogliamo lo schema concettuale e le considerazioni/vincoli finali.

1.2 Vincoli di dominio

- solo un parente di un paziente ricoverato può far visita ad un paziente ricoverato
- tutte le stanze interne appartenenti ad uno stesso reparto devono appartenere allo stesso ospedale
- un posto letto non può essere occupato se non è assegnato
- Alcuni allestimenti rendono obbligatorio l'assegnazione di posti (es: ricovero)
- un indirizzo appartiene ad una ed una sola tra le Entità Ospedale, Esterna e Persona
- Un vice primario può sostituire solo il suo primario (al momento dell'inserimento del dato) e quindi non può sostituire più di un Primario alla volta dato che è sempre il suo
- Un paziente può prenotare solo esami a cui appartiene
- Un medico può prescrivere esami solo ad un paziente di cui è responsabile
- un paziente non può essere ricoverato se è ricoverato in questo momento (non è finito il ricovero data fine ora fine)
- ogni reparto può avere al massimo un primario
- un ricovero deve essere effettuato in una stanza allestita appositamente
- un esame deve essere effettuato in una stanza allestita appositamente (laboratorio/ambulatorio)
- un dipendente deve aver finito di lavorare per poter cominciare un nuovo lavoro nello stesso reparto (data fine prima di data inizio)
- se un reparto è di pronto soccorso allora ci possono lavorare solamente medici o infermieri
- I medici e gli infermieri che prestano servizio in un reparto
- giorno settimanale può avere solo 7 valori che sono i giorni della settimana
- un dipendente può lavorare in più reparti ma solo dello stesso ospedale
- i turni di un dipendente non si devono sovrapporre
- un dipendente non deve superare le 8 ore di lavoro totali (ore turni)
- un dipendente non può avere un contratto di lavoro se ce n'è già uno in corso (data inizio , data fine)
- Lavoro(data inizio, data fine) devono essere assegnate solo ad un dipendente che abbia un contratto in corso in quel periodo (il contratto non può scadere prima della data fine del lavoro)
- un posto letto può essere assegnato solo ad un paziente che viene ricoverato nella stessa stanza in cui è presente quello specifico posto letto
- anzianità di servizio si incrementerà di 1 per ogni anno in cui il dipendente ha lavorato in quel ruolo (contratto)
- anzianità di servizio = Conseguimento(anni di esperienza) per quel titolo lavorativo (se esiste quindi ha già fatto esperienza con quel titolo) + anni passati con quel titolo lavorativo in azienda (quindi dall'inizio Contratto)
- Un vice primario può sostituire il proprio primario solo nel proprio turno di lavoro perchè se no come fa se non sta lavorando?
- Solo un vice primario può sostituire il primario e non il contrario
- un vice primario può sostituire il proprio primario solo se non lo sta già facendo (quindi la data fine deve essere passata)
- un paziente non può eseguire prenotazioni non dedicate a lui (in cui non è presente nella prescrizione)
- un paziente non può essere ricoverato per patologie che non ha
- un ricovero deve occupare un posto letto libero e non occupato
- se la tipologia di allestimento è stanza di ricovero allora i posti letto sono obbligatori
- un indirizzo deve corrispondere ad una ed una sola delle 3 entità collegate
- persona(username, password) sono unique e nullable

2. Progettazione logica

2.2 Vincoli di dominio



- solo un parente di un paziente ricoverato può far visita ad un paziente ricoverato
- tutte le stanze interne appartenenti ad uno stesso reparto devono appartenere allo stesso ospedale
- un posto letto non può essere occupato se non è assegnato
- Alcuni allestimenti rendono obbligatorio l'assegnazione di posti (es: ricovero)
- un indirizzo appartiene ad una ed una sola tra le Entità Ospedale, Esterna e Persona
- Un vice primario può sostituire solo il suo primario (al momento dell'inserimento del dato) e quindi non può sostituire più di un Primario alla volta dato che è sempre il suo
- Un paziente può prenotare solo esami a cui appartiene
- Un medico può prescrivere esami solo ad un paziente di cui è responsabile
- un paziente non può essere ricoverato se è ricoverato in questo momento (non è finito il ricovero data fine ora fine)
- ogni reparto può avere al massimo un primario
- un ricovero deve essere effettuato in una stanza allestita appositamente
- un esame deve essere effettuato in una stanza allestita appositamente (laboratorio/ambulatorio)
- un dipendente deve aver finito di lavorare per poter cominciare un nuovo lavoro nello stesso reparto (data fine prima di data inizio)
- se un reparto è di pronto soccorso allora ci possono lavorare solamente medici o infermieri
- I medici e gli infermieri che prestano servizio in un reparto
- giorno settimanale può avere solo 7 valori che sono i giorni della settimana
- un dipendente può lavorare in più reparti ma solo dello stesso ospedale
- i turni di un dipendente non si devono sovrapporre
- un dipendente non deve superare le 8 ore di lavoro totali (ore turni)
- un dipendente non può avere un contratto di lavoro se ce n'è già uno in corso (data inizio , data fine)
- Lavoro(data inizio, data fine) devono essere assegnate solo ad un dipendente che abbia un contratto in corso in quel periodo (il contratto non può scadere prima della data fine del lavoro)
- un posto letto può essere assegnato solo ad un paziente che viene ricoverato nella stessa stanza in cui è presente quello specifico posto letto
- anzianità di servizio si incrementerà di 1 per ogni anno in cui il dipendente ha lavorato in quel ruolo (contratto)
- anzianità di servizio = Conseguimento(anni di esperienza) per quel titolo lavorativo (se esiste quindi ha già fatto esperienza con quel titolo) + anni passati con quel titolo lavorativo in azienda (quindi dall'inizio Contratto)
- Un vice primario può sostituire il proprio primario solo nel proprio turno di lavoro perchè se no come fa se non sta lavorando?
- Solo un vice primario può sostituire il primario e non il contrario
- un vice primario può sostituire il proprio primario solo se non lo sta già facendo (quindi la data fine deve essere passata)
- un paziente non può eseguire prenotazioni non dedicate a lui (in cui non è presente nella prescrizione)
- un paziente non può essere ricoverato per patologie che non ha
- un ricovero deve occupare un posto letto libero e non occupato
- se la tipologia di allestimento è stanza di ricovero allora i posti letto sono obbligatori
- un indirizzo deve corrispondere ad una ed una sola delle 3 entità collegate
- persona(username, password) sono unique e nullable
- > una stanza può essere o interna o esterna
- > se ruolo di un dipendente è "Medico" allora l'associazione Per-Per è obbligatoria, se non lo è allora quell'associazione è proibita
- > un dipendente può essere o "Infermiere" o "Amministrativo" oppure "Medico"
- > Contratto(terminePrevisto) è obbligatorio se Contratto(tipologia) è "contratto indeterminato"
- > in Sostituzione e in Ricovero se è presente oraFine deve essere presente anche

dataFine e viceversa

2.3 Modello relazionale

TitoloLavorativo(nome, campo)

Specializzazione(nome, descrizioneBreve)

Persona(cf, nome, cognome, data di nascita, luogo di nascita, tel, email, via, nCiv, cap, username, password)

Paziente(tesseraSanitaria, cf, medico)

Prescrizione(numero, scadenza, urgenza, nota, medico, esame, paziente)

Esame(codice, descrizione, costoDaPrivato, costoConAssistenzaSanitaria, testoAvvertenze, tipologia)

Patologia(nome)

PostoLetto(id, stato, allestimentoRiempimento, stanzaRiempimento)

Allestimento(nome, tipologia)

Stanza(id)

Esterna(id, orariDiApertura, nTel, via, nCiv, cap)

Ospedale(codice, nome, via, nCiv, cap)

Turno(giornoSettimanale, oraInizio, oraFine)

Indirizzo(via, nCiv, cap, città, interno)

Dipendente(cf, esperienza, ruolo)

Contratto(dipendente, dataInizio, stipendio, percentualeAumentoAnnuo, oreLavoroSettimanali, terminePrevisto, tipologia, titoloLavorativo, dataFine)

Possiede(dipendente, specializzazione, dataConseguimento, dataScadenza)

Medico(cf)

VicePrimario(cf, primario)

Primario(cf)

Sostituzione(vicePrimario, Primario, dataInizio, oraInizio, terminePrevisto, dataFine, oraFine)

Prenotazione(prescrizione, paziente, giorno, dataEsame, oraEsame, stanza)

Ricovero(paziente, dataInizio, oraInizio, terminePrevisto, stanza, postoLetto, dataFine, oraFine)

Riempimento(allestimento, stanza)

Interna(ospedale, codice, piano, altezza, larghezza, profondità, ospedaleReparto, nomeReparto,
idGenericoStanza)

Reparto(ospedale, nome, tel, orarioDiVisita)

Lavoro(dipendente, ospedaleReparto, nomeReparto, dataInizio, terminePrevisto, dataFine)

Conseguito(dipendente, titoloLavorativo)

Presenza(paziente, patologia)

Obbiettivo(pazienteRicoverato, dataInizioRicovero, oraInizioRicovero, patologia)

Accordo(ospedale, stanzaEsterna)

Svolge(dipendente, ospedaleReparto, nomeReparto, dataInizio, giornoSettimanaleTurno,
oraInizioTurno, oraFineTurno)

SlotTemporale(dataesame, oraesame, stanza, occupato)

2.4 Codice SQL

```
CREATE TABLE TitoloLavorativo(  
    nome varchar(50) PRIMARY KEY,  
    campo varchar(50) NOT NULL  
);
```

```
CREATE TABLE Specializzazione(  
    nome varchar(50) PRIMARY KEY,  
    descrizioneBreve varchar(500) NOT NULL  
);
```

```
CREATE TABLE Persona(
    cf char(16) PRIMARY KEY,
    nome varchar(50) NOT NULL,
    cognome varchar(50) NOT NULL,
    dataDiNascita date NOT NULL CHECK(dataDiNascita > '1900-01-01'),
    luogoDiNascita varchar(30) NOT NULL,
    tel decimal(10,0) NOT NULL UNIQUE,
    email varchar(50) NOT NULL UNIQUE,
    via varchar(50) NOT NULL,
    nCiv decimal(4,0) NOT NULL,
    cap decimal(5,0) NOT NULL,
    username varchar(30) DEFAULT 'Guest',
    password varchar(30) DEFAULT 'Guest',
    UNIQUE(username, password),
    UNIQUE(nome, cognome, dataDiNascita)
);
```

```
CREATE TABLE Paziente(
    tesseraSanitaria decimal(20,0) PRIMARY KEY,
    cf char(16) NOT NULL UNIQUE,
    medico char(16)
);
```

```
CREATE TABLE Prescrizione(
    numero varchar(20) PRIMARY KEY,
    scadenza date CHECK(scadenza >= CURRENT_TIMESTAMP),
    urgenza decimal(1,0) NOT NULL,
    nota varchar(50),
    medico char(16) NOT NULL,
    esame varchar(20) NOT NULL,
    paziente decimal(20,0) NOT NULL
);
```

```
CREATE TABLE Esame(
    codice varchar(20) PRIMARY KEY,
    descrizione varchar(300) NOT NULL,
    costoDaPrivato decimal(7,2) NOT NULL,
    costoConAssistenzaSanitaria decimal(7,2) NOT NULL,
    testoAvvertenze varchar(1000) NOT NULL DEFAULT 'none',
    tipologia varchar(300) NOT NULL
);
```

```
CREATE TABLE Patologia(
    nome varchar(50) PRIMARY KEY
);
```

```
CREATE TABLE PostoLetto(
    id varchar(6) PRIMARY KEY,
    stato varchar(8) NOT NULL,
    allestimentoRiempimento varchar(100),
    stanzaRiempimento varchar(6)
);
```

```
CREATE TABLE Allestimento(
    nome varchar(100) PRIMARY KEY,
    tipologia varchar(50) NOT NULL
);
```


);

```
CREATE TABLE Stanza(  
    id varchar(6) PRIMARY KEY  
);
```

```
CREATE TABLE Esterna(  
    id varchar(6) PRIMARY KEY,  
    orariDiApertura varchar(100) NOT NULL,  
    nTel decimal(10,0) NOT NULL UNIQUE,  
    via varchar(50) NOT NULL,  
    nCiv decimal(4,0) NOT NULL,  
    cap decimal(5,0) NOT NULL  
);
```

```
CREATE TABLE Ospedale(  
    codice varchar(3) PRIMARY KEY,  
    nome varchar(100),  
    via varchar(50) NOT NULL,  
    nCiv decimal(4,0) NOT NULL,  
    cap decimal(5,0) NOT NULL  
);
```

```
CREATE TABLE Turno(  
    giornoSettimanale varchar(9),  
    oraInizio varchar(10),  
    oraFine varchar(10),  
    PRIMARY KEY(giornoSettimanale, oraInizio, oraFine)  
);
```

```
CREATE TABLE Indirizzo(  
    via varchar(50) NOT NULL,  
    nCiv decimal(4,0) NOT NULL,  
    cap decimal(5,0) NOT NULL,  
    città varchar(30) NOT NULL,  
    interno varchar(50),  
    PRIMARY KEY(via, nCiv, cap)  
);
```

```
CREATE TABLE Dipendente(  
    cf char(16) PRIMARY KEY,  
    esperienza decimal(2,0) NOT NULL DEFAULT 0,  
    ruolo varchar(100) NOT NULL  
);
```

```
CREATE TABLE Contratto(  
    dipendente char(16),  
    dataInizio date CHECK(dataInizio>=CURRENT_TIMESTAMP),  
    stipendio decimal(8,2) NOT NULL,  
    percentualeAumentoAnnuo decimal(4,2) NOT NULL,  
    oreLavoroSettimanali decimal(2,0) NOT NULL,  
    terminePrevisto date CHECK(terminePrevisto>=dataInizio),  
    tipologia varchar(50) NOT NULL,  
    titoloLavorativo varchar(100) NOT NULL,  
    dataFine date CHECK(dataFine>=dataInizio),  
    PRIMARY KEY(dipendente, dataInizio)  
);
```

```
CREATE TABLE Possiede(  
    dipendente char(16),  
    specializzazione varchar(100),  
    dataConseguimento date,  
    dataScadenza date,  
    PRIMARY KEY(dipendente, specializzazione, dataConseguimento)  
);
```

```
CREATE TABLE Medico(  
    cf char(16) PRIMARY KEY  
);
```

```
CREATE TABLE VicePrimario(  
    cf char(16) PRIMARY KEY,  
    primario char(16) NOT NULL  
);
```

```
CREATE TABLE Primario(  
    cf char(16) PRIMARY KEY  
);
```

```
CREATE TABLE Sostituzione(  
    vicePrimario char(16),  
    primario char(16),  
    dataInizio date CHECK(dataInizio>=CURRENT_TIMESTAMP),  
    oraInizio varchar(8),  
    terminePrevisto date CHECK(terminePrevisto>=dataInizio),  
    dataFine date CHECK(dataFine>=dataInizio),  
    oraFine varchar(8),  
    PRIMARY KEY(vicePrimario, primario, dataInizio, oraInizio)  
);
```

```
CREATE TABLE Prenotazione(  
    prescrizione varchar(20),  
    paziente decimal(20,0),  
    giorno date,  
    dataEsame date NOT NULL CHECK(dataEsame>=giorno),  
    oraEsame varchar(8) NOT NULL,  
    stanza varchar(6) NOT NULL,  
    PRIMARY KEY(prescrizione, paziente, giorno)  
);
```

```
CREATE TABLE Ricovero(  
    paziente decimal(20,0),  
    dataInizio date CHECK(dataInizio>=CURRENT_TIMESTAMP),  
    oraInizio varchar(8),  
    terminePrevisto date CHECK(terminePrevisto>=dataInizio),  
    stanza varchar(6) NOT NULL,  
    postoLetto varchar(6) NOT NULL,  
    dataFine date CHECK(dataFine>=dataInizio),  
    oraFine varchar(8),  
    PRIMARY KEY(paziente, dataInizio, oraInizio)  
);
```

```
CREATE TABLE Riempimento(  
    allestimento varchar(100),
```

```

        stanza varchar(6),
        PRIMARY KEY(allestimento, stanza)
    );

CREATE TABLE Interna(
    ospedale varchar(3),
    codice varchar(6),
    piano decimal(2,0) NOT NULL,
    altezza decimal(8,2) NOT NULL,
    larghezza decimal(8,2) NOT NULL,
    profondita decimal(8,2) NOT NULL,
    ospedaleReparto varchar(3) NOT NULL,
    nomeReparto varchar(100) NOT NULL,
    idGenericoStanza varchar(6) NOT NULL UNIQUE,
    PRIMARY KEY(ospedale, codice)
);

CREATE TABLE Reparto(
    ospedale varchar(3),
    nome varchar(100),
    tel decimal(10,0) NOT NULL UNIQUE,
    orarioDiVisita varchar(100) NOT NULL,
    PRIMARY KEY(ospedale, nome)
);

CREATE TABLE Lavoro(
    dipendente char(16),
    ospedaleReparto varchar(3),
    nomeReparto varchar(100),
    dataInizio date CHECK(dataInizio>=CURRENT_TIMESTAMP),
    terminePrevisto date CHECK(terminePrevisto>=dataInizio),
    dataFine date CHECK(dataFine>=dataInizio),
    PRIMARY KEY(dipendente, ospedaleReparto, nomeReparto, dataInizio)
);

CREATE TABLE Conseguito(
    dipendente char(16),
    titoloLavorativo varchar(50),
    PRIMARY KEY(dipendente, titoloLavorativo)
);

CREATE TABLE Presenza(
    paziente decimal(20,0),
    patologia varchar(50),
    PRIMARY KEY(paziente, patologia)
);

CREATE TABLE Obbiettivo(
    pazienteRicoverato decimal(20,0),
    dataInizioRicovero date,
    oraInizioRicovero varchar(8),
    patologia varchar(50) NOT NULL,
    PRIMARY KEY(pazienteRicoverato, dataInizioRicovero, oraInizioRicovero)
);

CREATE TABLE Accordo(
    ospedale varchar(3),

```

```

        stanzaEsterna varchar(6),
        PRIMARY KEY(ospedale, stanzaEsterna)
    );

CREATE TABLE Svolge(
    dipendente char(16),
    ospedaleReparto varchar(3),
    nomeReparto varchar(100),
    dataInizio date,
    giornoSettimanaleTurno varchar(9),
    oraInizioTurno varchar(8),
    oraFineTurno varchar(8),
    PRIMARY KEY(dipendente, ospedaleReparto, nomeReparto, dataInizio,
giornoSettimanaleTurno, oraInizioTurno, oraFineTurno)
);

CREATE TABLE slottemporale(
    dataEsame date NOT NULL,
    oraEsame varchar(11) NOT NULL,
    stanza varchar(6) NOT NULL,
    occupato varchar(1) NOT NULL,
    PRIMARY KEY(dataesame, oraesame, stanza)
);

```

```

ALTER TABLE persona ADD FOREIGN KEY (via,nciv,cap) REFERENCES
indirizzo(via,nciv,cap) ON UPDATE CASCADE;

```

```

ALTER TABLE paziente ADD FOREIGN KEY (cf) REFERENCES persona(cf) ON
UPDATE CASCADE;

```

```

ALTER TABLE paziente ADD FOREIGN KEY (medico) REFERENCES medico(cf) ON
UPDATE CASCADE;

```

```

ALTER TABLE prescrizione ADD FOREIGN KEY (medico) REFERENCES medico(cf)
ON UPDATE CASCADE;

```

```

ALTER TABLE prescrizione ADD FOREIGN KEY (esame) REFERENCES esame(codice)
ON UPDATE CASCADE;

```

```

ALTER TABLE prescrizione ADD FOREIGN KEY (paziente) REFERENCES
paziente(tesserasanitaria) ON UPDATE CASCADE;

```

```

ALTER TABLE postoletto ADD FOREIGN KEY (allestimentoriempimento,
stanzariempimento) REFERENCES riempimento(allestimento, stanza) ON UPDATE
CASCADE;

```

```

ALTER TABLE esterna ADD FOREIGN KEY (id) REFERENCES stanza(id) ON
UPDATE CASCADE;

```

```

ALTER TABLE esterna ADD FOREIGN KEY (via, nciv, cap) REFERENCES
indirizzo(via, nciv, cap) ON UPDATE CASCADE;

```

ALTER TABLE ospedale ADD FOREIGN KEY (via, nciv, cap) REFERENCES indirizzo(via, nciv, cap) ON UPDATE CASCADE;

ALTER TABLE dipendente ADD FOREIGN KEY (cf) REFERENCES persona(cf) ON UPDATE CASCADE;

ALTER TABLE contratto ADD FOREIGN KEY (dipendente) REFERENCES dipendente(cf) ON UPDATE CASCADE;

ALTER TABLE contratto ADD FOREIGN KEY (titololavorativo) REFERENCES titololavorativo(nome) ON UPDATE CASCADE;

ALTER TABLE possiede ADD FOREIGN KEY (dipendente) REFERENCES dipendente(cf) ON UPDATE CASCADE;

ALTER TABLE possiede ADD FOREIGN KEY (specializzazione) REFERENCES specializzazione(nome) ON UPDATE CASCADE;

ALTER TABLE medico ADD FOREIGN KEY (cf) REFERENCES dipendente(cf) ON UPDATE CASCADE;

ALTER TABLE viceprimario ADD FOREIGN KEY (cf) REFERENCES dipendente(cf) ON UPDATE CASCADE;

ALTER TABLE viceprimario ADD FOREIGN KEY (primario) REFERENCES primario(cf) ON UPDATE CASCADE;

ALTER TABLE primario ADD FOREIGN KEY (cf) REFERENCES dipendente(cf) ON UPDATE CASCADE;

ALTER TABLE sostituzione ADD FOREIGN KEY (viceprimario) REFERENCES viceprimario(cf) ON UPDATE CASCADE;

ALTER TABLE sostituzione ADD FOREIGN KEY (primario) REFERENCES primario(cf) ON UPDATE CASCADE;

ALTER TABLE prenotazione ADD FOREIGN KEY (prescrizione) REFERENCES prescrizione(numero) ON UPDATE NO ACTION;

ALTER TABLE prenotazione ADD FOREIGN KEY (paziente) REFERENCES paziente(tesserasanitaria) ON UPDATE CASCADE;

ALTER TABLE prenotazione ADD FOREIGN KEY (dataesame, oraesame, stanza) REFERENCES slotttemporale(dataesame, oraesame, stanza) ON UPDATE CASCADE;

ALTER TABLE ricovero ADD FOREIGN KEY (paziente) REFERENCES paziente(tesserasanitaria) ON UPDATE CASCADE;

ALTER TABLE ricovero ADD FOREIGN KEY (stanza) REFERENCES stanza(id) ON UPDATE CASCADE;

ALTER TABLE ricovero ADD FOREIGN KEY (postoletto) REFERENCES postoletto(id) ON UPDATE CASCADE;

ALTER TABLE riempimento ADD FOREIGN KEY (allestimento) REFERENCES allestimento(nome) ON UPDATE CASCADE;

*ALTER TABLE riempimento ADD FOREIGN KEY (stanza) REFERENCES stanza(id)
ON UPDATE CASCADE;*

*ALTER TABLE interna ADD FOREIGN KEY (ospedale) REFERENCES
ospedale(codice) ON UPDATE CASCADE;*

*ALTER TABLE interna ADD FOREIGN KEY (idgenericostanza) REFERENCES
stanza(id) ON UPDATE CASCADE;*

*ALTER TABLE interna ADD FOREIGN KEY (ospedalereparto, nomereparto)
REFERENCES reparto(ospedale, nome) ON UPDATE CASCADE;*

*ALTER TABLE reparto ADD FOREIGN KEY (ospedale) REFERENCES
ospedale(codice) ON UPDATE CASCADE;*

*ALTER TABLE lavoro ADD FOREIGN KEY (dipendente) REFERENCES dipendente(cf)
ON UPDATE CASCADE;*

*ALTER TABLE lavoro ADD FOREIGN KEY (ospedalereparto, nomereparto)
REFERENCES reparto(ospedale, nome) ON UPDATE CASCADE;*

*ALTER TABLE conseguito ADD FOREIGN KEY (dipendente) REFERENCES
dipendente(cf) ON UPDATE CASCADE;*

*ALTER TABLE conseguito ADD FOREIGN KEY (titololavorativo) REFERENCES
titololavorativo(nome) ON UPDATE CASCADE;*

*ALTER TABLE presenza ADD FOREIGN KEY (paziente) REFERENCES
paziente(tesserasanitaria) ON UPDATE CASCADE;*

*ALTER TABLE presenza ADD FOREIGN KEY (patologia) REFERENCES
patologia(nome) ON UPDATE CASCADE;*

*ALTER TABLE obbiettivo ADD FOREIGN KEY (patologia) REFERENCES
patologia(nome) ON UPDATE CASCADE;*

*ALTER TABLE obbiettivo ADD FOREIGN KEY (pazientericoverato, datainizioricovero,
orainizioricovero) REFERENCES ricovero(paziente, datainizio, orainizio) ON UPDATE
CASCADE;*

*ALTER TABLE accordo ADD FOREIGN KEY (ospedale) REFERENCES
ospedale(codice) ON UPDATE CASCADE;*

*ALTER TABLE accordo ADD FOREIGN KEY (stanzaesterna) REFERENCES esterna(id)
ON UPDATE CASCADE;*

*ALTER TABLE svolge ADD FOREIGN KEY (dipendente, ospedalereparto, nomereparto,
datainizio) REFERENCES lavoro(dipendente, ospedalereparto, nomereparto, datainizio)
ON UPDATE CASCADE;*

*ALTER TABLE svolge ADD FOREIGN KEY (giornosettimanaleturno, orainizioturno,
orafineturno) REFERENCES turno(giornosettimanale, orainizio, orafine) ON UPDATE
CASCADE;*

*ALTER TABLE slottemporale ADD FOREIGN KEY (stanza) REFERENCES stanza(id)
ON UPDATE CASCADE;*

3. Progettazione del sito

Select a table

ospedale select

new guest prescribe an exam book an exam

SHORTCUTS

SELEZIONA LA TABELLA DA GESTIRE

The ER diagram illustrates the following entities and relationships:

- Specializzazione** (Entity): Attributes include *nome* and *descrizioneBreve*.
- Spe-Pos** (Relationship): Connects **Specializzazione** and **Possiede** with cardinalities (0,n) and (1,1).
- Possiede** (Entity): Attributes include *dataConsegimento* and *dataScadenza*.
- Per-Pos** (Relationship): Connects **Possiede** and **Titolo lavorativo** with cardinalities (1,1) and (0,n).
- Titolo lavorativo** (Entity): Attributes include *Nome*, *campo*, *anniEsperienza*, *oreLavoroSettimanale*, *percentualeAumentoAnnuo*, *stipendio*, and *dataInizio*.
- Conseguito** (Relationship): Connects **Titolo lavorativo** and **Possiede** with cardinalities (0,n) and (0,n).
- Composizione** (Relationship): Connects **Titolo lavorativo** and **Per-Pos** with cardinalities (0,n) and (1,1).

Tabella: ospedale

codice	nome	via FK	nciv FK	cap FK		
01	Borgo Trento	Giuseppe Randagio	3	37139	delete	update
02	Borgo Roma	Dal Grande Osor	21	37130	delete	update
03	Lasciavera	Farnesa	1	35110	delete	update
che	naninani	Farnesa	1	35110	delete	update
hla	shadow clone Jutzu	Farnesa	1	35110	delete	update
zla	parson project	Luigi Dosso	53	37139	delete	update
insert						

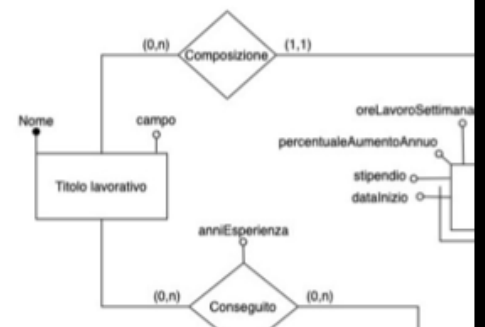
Go Back

Back to Home Page

INSERISCI NUOVO VALORE

ELIMINA RIGA

AGGIORNA I VALORI



Dopo aver cliccato su “insert”, inserire i valori dall’alto verso il basso

Tabella: ospedale

Nome	Inserimento	Tipologia dato	
codice	<input type="text"/>	character varying 3 ----- Not nullable	
nome	<input type="text"/>	character varying 100 ----- Nullable	
via	<input type="text"/>	character varying 50 ----- Not nullable	<input type="button" value="Jump"/>
nciv	<input type="text"/>	numeric 4 ----- Not nullable	<input type="button" value="Jump"/>
cap	<input type="text"/>	numeric 5 ----- Not nullable	<input type="button" value="Jump"/>
<input type="button" value="submit"/>	<div>0 20027 35110 37130 37138 37139</div>		
<input type="button" value="Go Back"/>	<input type="button" value="set"/>		

Durante l’inserimento di valori che presentano un vincolo di chiave esterna verranno fornite le opzioni possibili tra cui scegliere, se l’opzione non è presente usare il pulsante di “Jump” per andare a gestire la tabella da cui vengono presi i valori possibili

Arancio -> chiave primaria

Azzurro -> chiave esterna

Nome del database PostgreSQL: hospitals
Nome della cartella sotto htdocs: project

Per effettuare il login al sistema:

Nome Utente: unimi
Password: unimi