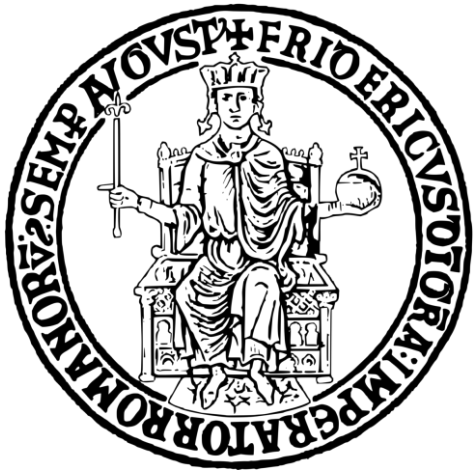


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Corso di Laurea in

INFORMATICA

Documentazione basi di dati

Progetto “Voyager”

A cura di
Alessandro Preziosi
Alessandro Ferro

Anno accademico
2019/2020

INDICE

Specifiche

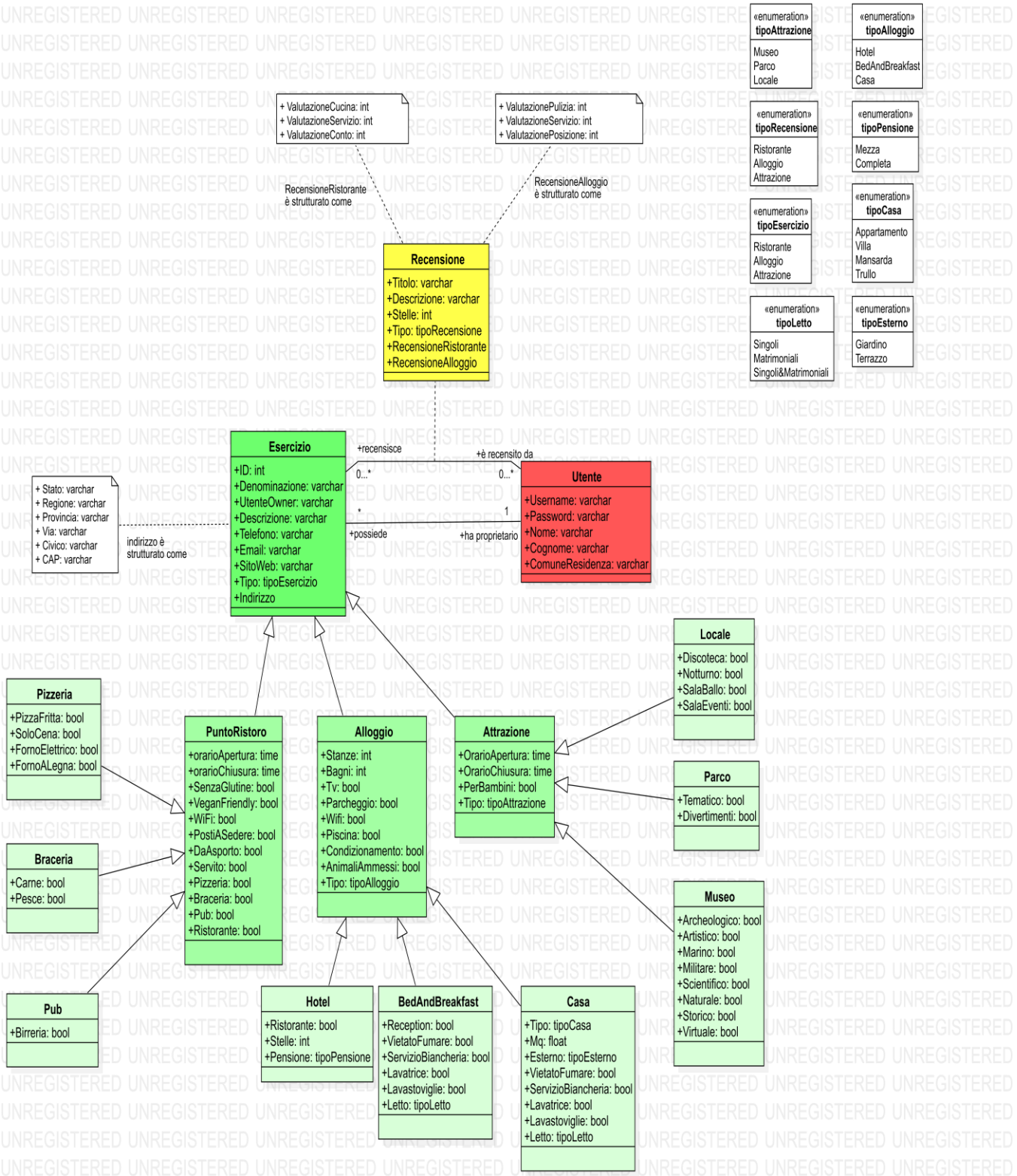
Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo in Java Swing, per la gestione di un sistema di recensioni turistiche, tipo TripAdvisor. In particolare, il sistema deve permettere di gestire tre categorie Ristoranti, Alloggi e Attrazioni. Ognuna di queste tre categorie va ulteriormente raffinata (ad esempio un ristorante può essere sia "Pizzeria" che "Braceria"). Il sistema deve permettere anche di gestire le recensioni, forzando il vincolo che ogni utente possa pubblicare al più UNA recensione per elemento visitato.

Introduzione

Il programma realizzato permette di far inserire nuovi esercizi commerciali all’utente, previa iscrizione dello stesso, e di gestirne le recensioni che aiutano a descrivere l’esperienza dei visitatori anche cambiando criteri di valutazione in base all’attività commerciale da valutare.

L’applicativo desktop utilizza il DBMS “PostgreSQL”, il cui server si trova in remoto sulla piattaforma Amazon Web Services.

Class diagram non ristrutturato



Si richiede di gestire tre categorie: Ristoranti, Alloggi e Attrazioni, ognuna delle quali va ulteriormente raffinata.

Per rispondere a tale esigenza si è pensato di creare una classe “Esercizio” che si specializza in PuntoRistoro (il cui nome differisce dalla specifica per distinguerlo dalla sua sotto-specializzazione), Alloggio e Attrazione.

Ognuna di esse si raffina ulteriormente in questo modo:

- PuntoRistoro: Pizzeria, Braceria, Pub, Ristorante (la cui classe non è implementata in quanto i suoi attributi non differiscono dalla classe che specializzerebbe);
- Alloggio: Hotel, BedAndBreakfast, Casa;
- Attrazione: Locale, Parco, Museo.

Modifiche di ristrutturazione

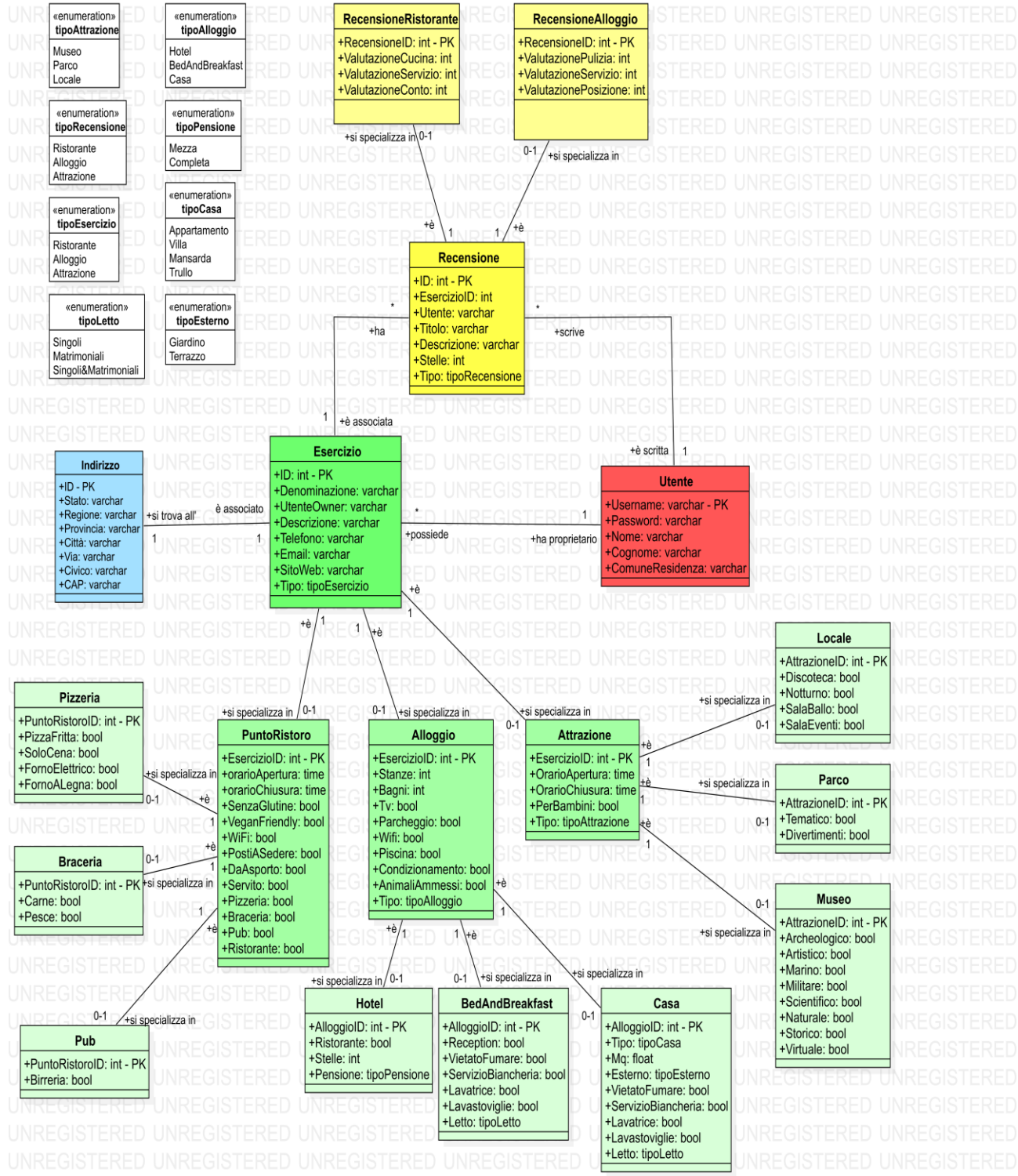
I) La prima modifica di ristrutturazione effettuata riguarda le specializzazioni degli esercizi commerciali: esse vengono tramutate in associazioni 1 a [0-1] dal momento che ognuna di esse rappresenta una specializzazione del singolo esercizio in oggetto.
Le specializzazioni di PuntoRistoro sono *total overlapping* per rispettare le specifiche fornite, mentre di Alloggio e Attrazione sono *total disjoint.*

II) L’associazione N a N “Recensione” tra Esercizio e Utente è stata tramutata in due associazioni 1 a N grazie alla classe “Recensione”.

III) La nuova classe Recensione è stata specializzata in “RecensioneRistorante” e “RecensioneAlloggio” per permettere al recensore di esprimere al meglio la propria opinione. Se ciò che si sta recensendo non è né un Alloggio né un Ristorante (e dunque un’attrazione), non vi è una specializzazione, ed è dunque un partial disjoint.

IV) L’attributo strutturato “indirizzo” presente nella classe Esercizio è stato ristrutturato in una classe a parte collegata con un’associazione 1 a 1 con la classe Esercizio.

Class diagram ristrutturato



Dettagli sulle classi

Utente

Questa classe ha lo scopo di modellare una persona.

Attributi:

- username (*primary key);*
- Nome: *varchar*;
- Cognome: *varchar*;
- ComuneResidenza: *varchar*;
- Password: *varchar*.

Esercizio

Questa classe è una rappresentazione di un esercizio commerciale, contenente attributi plausibilmente comuni a qualunque tipologia di esercizio.

Attributi:

- ID: *int (primary key);*
- denominazione: *varchar;*
- utenteOwner: *varchar (foreign key verso Utente);*
- Descrizione: *varchar;*
- Telefono: *varchar;*
- e-mail: *varchar;*
- sitoWeb: *varchar;*
- tipo: *TipoEsercizio (indica la sotto-tipologia di tale esercizio);*
- id_indirizzo: *int (foreign key verso Indirizzo).*

PuntoRistoro

È la rappresentazione di un generico esercizio adibito alla ristorazione.

Tra i vari attributi ne presenta quattro booleani (Pizzeria, Braceria, Pub, Ristorante) che stanno a specificare il tipo (o i tipi) del punto ristorazione.

Attributi:

- esercizioID: *int (primary key e foreign key verso Esercizio)*
- orarioApertura: *time*;

- orarioChiusura: *time*;
- senzaGlutine: *boolean*;
- veganFriendly: *boolean*;
- WiFi: *boolean*;
- postiASedere: *boolean*;
- daAsporto: *boolean*;
- servito: *boolean*;
- pizzeria: *boolean*;
- braceria: *boolean*;
- pub: *boolean*;
- ristorante: *boolean*.

Pizzeria

È una rappresentazione di una pizzeria del mondo reale.

Attributi:

- puntoRistoroId: int (primary key - foreign key verso PuntoRistoro);
- pizzaFritta: *boolean*;
- soloCena: *boolean*;
- fornoElettrico: *boolean*;
- fornoALegna: *boolean*.

Braceria

È una rappresentazione di una braceria del mondo reale.

Attrributi:

- puntoRistoroId: *int* (primary key - foreign key verso PuntoRistoro);
- carne: *boolean*;
- pesce: *boolean*.

Pub

È una rappresentazione di un pub del mondo reale.

Attributi:

- puntoRistoroId: *int* (rimary key - foreign key verso PuntoRistoro);
- birreria: *boolean*.

Alloggio

Rappresentazione di un generico alloggio, il quale rappresenta la generalizzazione di diversi tipi di alloggi esistenti. L’attributo “Tipo” indica in quale di essi esso si specializza.

Attributi:

- esercizioId: *int* (primary key e foreign key verso Esercizio);
- stanze: *int*;
- bagni: *int*;
- tv: *boolean*;
- parcheggio: *boolean*;
- WiFi: *boolean*;
- piscina: *boolean*;
- condizionamento: *boolean*;
- animaliAmmessi: *boolean*;
- tipo: *ENUM(‘Hotel’ , ’BedAndBreakfast’ , ’Casa’)*.

Hotel

È una rappresentazione di un Hotel del mondo reale.

Attributi:

- alloggioId: *int (primary key e foreign key verso Alloggio)*;
- ristorante: *boolean*;
- stelle: *int*;
- pensione: *ENUM(‘ mezza ’ , ‘ completa ’).*

BedAndBreakfast

È una rappresentazione di un BedAndBreakfast del mondo reale.

Attributi:

- alloggioId: *int* (primary key - foreign key verso Alloggio);
- reception: *boolean*;
- vietatoFumare: *boolean*;
- lavatrice: *boolean*;
- lavastoviglie: *boolean*;
- letto: *ENUM(‘ Singolo ’ , ‘ Matrimoniale ’ , ‘ Singolo&Matrimoniale ’)*.

Casa

Rappresenta una casa del mondo reale.

Attributi:

- alloggioId: *int (primary key e foreign key verso Alloggio)*;
- tipo: *ENUM(‘ Appartamento ’ , ‘ Villa ‘ , ‘ Mansarda ‘ , ‘ Trullo ‘)*;
- mq: *decimal*;
- esterno: *ENUM(‘ Giardino ‘ , ‘ Terrazzo ‘)*;
- vietatoFumare: *boolean*;
- servizioBiancheria: *boolean*;
- lavatrice: *boolean*;
- lavastoviglie: *boolean*;
- letto: *ENUM(‘ Singolo ‘ , ‘ Matrimoniale ‘ , ‘ Singolo&Matrimoniale ‘)*.

Attrazione

Rappresentazione di una generica attrazione ricreativa.

Attributi:

- esercizioId: *int (primary key e foreign key verso Esercizio)*;
- orarioApertura: *time*;
- orarioChiusura: *time*;
- tipo: ENUM (‘ Museo ‘ , ‘ Parco ‘ , ‘ Locale ‘).

Locale

Rappresenta un locale ludico.

Attributi:

- attrazioneId: *int (primary key e foreign key verso Attrazione)*;
- discoteca: *boolean*;
- notturno: *boolean*;
- salaBallo: *boolean*;
- salaEventi: *boolean*.

Parco

È una rappresentazione di un parco divertimenti.

Attributi:

- attrazioneId: *int (primary key e foreign key verso Attrazione);*
- tematico: *boolean*;
- divertimenti: *boolean*.

Museo

Rappresenta un museo e le sue varie tipologie.

Attributi:

- attrazioneID *int (primary key – foreign key verso Attrazione)*;
- archeologico: *boolean*;
- artistico: *boolean*;
- marino: *boolean*;
- militare: *boolean*;
- scientifico: *boolean*;
- naturale: *boolean*;
- virtuale: *boolean*.

Recensione

Descrive una recensione per un esercizio commerciale.

Attributi:

- ID: *int (primary key)*;
- esercizioID: *int (foreign key verso Esercizio)*;
- utente: *varchar (foreign key verso la classe Utente)*;
- titolo: *varchar*;
- descrizione: *varchar*;
- stelle: *int*;
- tipo: *ENUM(‘ Ristorante ’ , ’ Alloggio ’).*

Indirizzo

È una classe che contiene attributi atti a conservare informazioni per descrivere un Indirizzo.

Attributi:

- ID: *int* (primary key);
- Stato: *varchar*;
- Regione: *varchar*;
- Provincia: *varchar*;
- Città: *varchar*;
- Via: *varchar*;
- Civico: *varchar*;
- CAP: *varchar;*

Dettagli sulle associazioni

- **Associazione Esercizio - Indirizzo**
  L’associazione tra Esercizio e Indirizzo è un’associazione 1 a 1. L’esercizio contiene una foreign key verso l’ID (auto increment) dell’indirizzo.
  Un esercizio si trova ad un indirizzo e un indirizzo è associato un esercizio
- **Associazione Esercizio - PuntoRistoro**
  L’associazione tra Esercizio e PuntoRistoro è di tipo 1 a [0-1] in quanto l’Esercizio può essere anche di un tipo diverso oltre a quello di PuntoRistoro.
  Un esercizio si specializza in un punto ristoro e un punto ristoro è un esercizio.
- **Associazione Esercizio - Alloggio**
  L’associazione tra Esercizio e Alloggio è di tipo 1 a [0-1] in quanto l’Esercizio può essere anche di un tipo diverso oltre a quello di Alloggio.
  Un esercizio si specializza in un alloggio e un alloggio è un esercizio.
- **Associazione Esercizio - Attrazione**
  L’associazione tra Esercizio e Attrazione è di tipo 1 a [0-1] in quanto l’Esercizio può essere anche di un tipo diverso oltre a quello di Attrazione. Un esercizio si specializza in un’attrazione e un’attrazione è un esercizio.
- **Associazione PuntoRistoro – Pizzeria**
  Si tratta di un’associazione 1 a [0-1]. Un punto ristoro si può specializzare in pizzeria e una pizzeria è un punto ristoro.
- **Associazione PuntoRistoro – Braceria**
  Si tratta di un’associazione 1 a [0-1]. Un punto ristoro si può specializzare in braceria e una braceria è un punto ristoro.

- **Associazione PuntoRistoro – Pub**
  Si tratta di un’associazione 1 a [0-1]. Un punto ristoro si può specializzare in pub e un pub è un punto ristoro.
- **Associazione Alloggio – Hotel**
  È un’associazione 1 a [0-1] in quanto un hotel è un alloggio e un alloggio si può specializzare in hotel.
- **Associazione Alloggio – BedAndBreakfast**
  È un’associazione 1 a [0-1] in quanto un bed and breakfast è un alloggio e un alloggio si può specializzare in bed and breakfast.
- **Associazione Alloggio – Casa**
  È un’associazione 1 a [0-1] in quanto una casa è un alloggio e un alloggio si può specializzare in casa.
- **Associazione Attrazione – Locale**
  Un locale è una e una sola attrazione e un’attrazione può essere al più un locale, pertanto l’associazione è di tipo 1 a [0-1].
- **Associazione Attrazione – Parco**
  Un parco è una e una sola attrazione e un’attrazione può essere al più un parco, pertanto l’associazione è di tipo 1 a [0-1].
- **Associazione Attrazione – Museo**
  Un museo è una e una sola attrazione e un’attrazione può essere al più un museo, pertanto l’associazione è di tipo 1 a [0-1].
- **Associazione Esercizio - Utente**
  Un utente iscritto all’applicativo ha la facoltà di inserire uno o più esercizi, mentre un esercizio ha uno e un solo proprietario. L’associazione ne risulta dunque essere 1 a N.
- **Associazione Utente – Recensione**
  L’associazione è di tipo 1 a N in quanto un Utente può scrivere N recensioni all’interno dell’applicativo, mentre una recensione è associata a un e un solo utente.
- **Associazione Esercizio – Recensione**
  Un esercizio può avere un numero indefinito di recensioni, ma una recensione è associata a un solo esercizio. L’associazione è di tipo 1 a N.

- **Associazione Recensione – RecensioneRistorante**
  L’associazione è 1 a [0-1] poiché una recensione può essere specializzata in una RecensioneRistorante se ciò che si sta recensendo è un ristorante.
- **Associazione Recensione - RecensioneAlloggio**
  L’associazione è 1 a [0-1] poiché una recensione può essere specializzata in una RecensioneAlloggio se ciò che si sta recensendo è un alloggio.

Dizionario dei vincoli

Vincolo	Regola
ProprietarioFK	Forza la FK *utenteOwner* in Esercizio ad essere associata ad un proprietario
IndirizzoFK	Forza la FK ID_Indirizzo ad essere associata a un indirizzo
StelleValide	Forza l’attributo rappresentante le stelle di una Recensione ad assumere un valore valido (tra 1 e 5, estremi compresi)
UtenteFK	Forza la FK utente in Recensione ad essere associata ad un autore
EsercizioFK_recensione	Forza la FK EsercizioID in Recensione ad essere associata ad un esercizio.
RecensioneUnica	Forza l’utente a poter inserire al più una recensione per ogni esercizio commerciale
RecensioneFK_ristorante	Forza la FK RecensioneID presente in RecensioneRistorante ad essere associata ad una recensione, essendone sua specializzazione
RecensioneFK_alloggio	Forza la FK recensioneID presente in RecensionaAlloggio ad essere associata ad una recensione, essendone sua specializzazione
EsercizioFK_alloggio	Forza la FK esercizioID in Alloggio ad essere associata ad un Esercizio
AlloggioFK_hotel	Forza la FK alloggioID in Hotel ad essere associata ad un Alloggio.
AlloggioFK_BB	Forza la FK alloggioID in BedAndBreakfast ad essere associata ad un Alloggio.
AlloggioFK_casa	Forza la FK alloggioID in Casa ad essere associata ad un Alloggio.
EsercizioFK_attrazione	Forza la FK esercizioID in Attrazione ad essere collegata a un Esercizio
AttrazioneFK_museo	Forza la FK attrazioneID in Museo ad essere associata ad un’ Attrazione
AttrazioneFK_parco	Forza la FK attrazioneID in Parco ad essere associata ad un’ Attrazione
AttrazioneFK_locale	Forza la FK attrazioneID in Locale ad essere associata ad un’ Attrazione
EsercizioFK_puntoristoro	Forza la FK esercizioID in PuntoRistoro ad essere collegata a un Esercizio
PuntoRistoroFK_pizzeria	Forza la FK puntoRistoroID in Pizzeria ad essere collegata a un PuntoRistoro
PuntoRistoroFK_braceria	Forza la FK puntoRistoroID in Braceria ad essere collegata a un PuntoRistoro

PuntoRistoroFK_pub	Forza la FK puntoRistoroID in Pub ad essere collegata a un PuntoRistoro

Schema logico

Nel seguente paragrafo gli attributi in **grassetto** stanno ad indicare una primary key, mentre quelli sottolineati individuano le foreign key.

Utente (**Username,** Password, Nome, Cognome, ComuneResidenza)

Esercizio (**ID,** utenteOwner, IDindirizzo, Denominazione, Descrizione, Telefono, Email, SitoWeb, Tipo)

Indirizzo (**ID,** Stato, Regione, Provincia, Città, Via, Civico, CAP)

Recensione (**ID,** Utente, EsercizioID, Titolo, Descrizione, Stelle, Tipo)

RecensioneRistorante (**RecensioneID**, ValutazioneCucina, ValutazioneMenu, ValutazioneServizio, ValutazioneConto)

RecensioneAlloggio (**RecensioneID**, ValutazionePulizia, ValutazioneServizio, ValutazionePrezzo, ValutazionePosizione, ValutazioneCibo)

Alloggio (**EsercizioID**, Stanze, Bagni, TV, Parcheggio, WiFi, Piscina, Condizionamento, AnimaliAmmessi, Tipo)

Hotel (**AlloggioID**, Ristorante, Stelle, Pensione)

BedAndBreakfast (**AlloggioID**, Reception, VietatoFumare, ServizioBiancheria, Lavatrice, Lavastoviglie, Letto)

Casa (**AlloggioID**, Tipo, Mq, Esterno, VietatoFumare, ServizioBiancheria, Lavatrice, Lavastoviglie, Leto)

Attrazione (**EsercizioID**, OrarioApertura, OrarioChiusura, PerBambini, Tipo)

Museo (**AttrazioneID**, Archeologico, Artistico, Marino, Militre, Scientifico, Storico, Naturale, Virtuale)

Parco (**AttrazioneID**, Tematico, Divertimenti)

Locale (**AttrazioneID,** Discoteca, Notturno, SalaBallo, SalaEventi)

PuntoRistoro (**EsercizioID**, OrarioApertura, OrarioChiusura, SenzaGlutine, VeganFriendly, WiFi, PostiASedere, DaAsporto, Servito, Pizzeria, Braceria, Pub, Ristorante)

Pizzeria (**PuntoRistoroID**, PizzaFritta, SoloCena, FornoElettrico, FornoLegna)

Braceria (**PuntoRistoroID**, Carne, Pesce)

Pub (**PuntoRistoroID**, Birreria)

Codice SQL (Oracle-compatible)

```
CREATE TABLE Utente(
	Username VARCHAR(16) PRIMARY KEY,
	Password VARCHAR(64) NOT NULL,
	Nome VARCHAR(32) NOT NULL,
	Cognome VARCHAR(32) NOT NULL,
	ComuneResidenza VARCHAR(64) NOT NULL
);

CREATE TABLE Esercizio(
	ID SERIAL PRIMARY KEY,
	Denominazione VARCHAR(64) NOT NULL,
	UtenteOwner VARCHAR(16),
	Descrizione VARCHAR(4096),
	Telefono VARCHAR(16),
	Email VARCHAR(32),
	SitoWeb VARCHAR(32),
	Tipo ENUM('Ristorante','Alloggio','Attrazione') NOT NULL;
	ID_indirizzo INT NOT NULL,
	CONSTRAINT ProprietarioFK FOREIGN KEY(UtenteOwner) REFERENCES
Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,
	CONSTRAINT IndirizzoFK FOREIGN KEY(ID_indirizzo) REFERENCES
Indirizzo(ID) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE Indirizzo(
	ID SERIAL PRIMARY KEY,
	Stato VARCHAR(32) NOT NULL,
	Regione VARCHAR(32) NOT NULL,
	Provincia VARCHAR(32) NOT NULL,
	Citta VARCHAR(32) NOT NULL,
	Via VARCHAR(32) NOT NULL,
	Civico VARCHAR(6) NOT NULL,
	CAP VARCHAR(8) NOT NULL
);

CREATE TABLE Recensione(
	ID SERIAL PRIMARY KEY,
	EsercizioID INT,
	Utente VARCHAR(16),
	Titolo VARCHAR(32) NOT NULL,
	Descrizione VARCHAR(2048) NOT NULL,
	Stelle INT NOT NULL,
	Tipo ENUM ('Ristorante','Alloggio') NOT NULL,
	CONSTRAINT StelleValide CHECK(Stelle BETWEEN 1 AND 5),
	CONSTRAINT UtenteFK FOREIGN KEY(Utente) REFERENCES
Utente(Username) ON DELETE CASCADE ON UPDATE NO ACTION,
	CONSTRAINT EsercizioFK_recensione FOREIGN KEY(EsercizioID)
REFERENCES Esercizio(ID) ON DELETE CASCADE ON UPDATE NO ACTION,
	CONSTRAINT RecensioneUnica UNIQUE(Utente,EsercizioID)
```

```
);

CREATE TABLE RecensioneRistorante(
	RecensioneID INT PRIMARY KEY,
	ValutazioneCucina INT NOT NULL CHECK (ValutazioneCucina BETWEEN
1 AND 5),
	ValutazioneServizio INT NOT NULL CHECK (ValutazioneServizio
BETWEEN 1 AND 5),
	ValutazioneConto INT NOT NULL CHECK (ValutazioneConto BETWEEN 1
AND 5),
	CONSTRAINT RecensioneFK_ristorante FOREIGN KEY(RecensioneID)
REFERENCES Recensione(ID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE RecensioneAlloggio(
	RecensioneID INT PRIMARY KEY,
	ValutazionePulizia INT CHECK (ValutazionePulizia BETWEEN 1 AND 5),
	ValutazioneServizio INT CHECK (ValutazioneServizio BETWEEN 1 AND
5),
	ValutazionePosizione INT CHECK (ValutazionePosizione BETWEEN 1 AND
5),
	CONSTRAINT RecensioneFK_alloggio FOREIGN KEY(RecensioneID)
REFERENCES Recensione(ID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Alloggio(
	EsercizioID INT PRIMARY KEY,
	Stanze INT,
	Bagni INT,
	TV BOOLEAN,
	Parcheggio BOOLEAN,
	WiFi BOOLEAN,
	Piscina BOOLEAN,
	Condizionamento BOOLEAN,
	AnimaliAmmessi BOOLEAN,
	Tipo ENUM ('Hotel','BedAndBreakfast','Casa') NOT NULL;
	CONSTRAINT EsercizioFK_alloggio FOREIGN KEY(EsercizioID)
REFERENCES Esercizio(ID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Hotel(
	AlloggioID INT PRIMARY KEY,
	Ristorante BOOLEAN,
	Stelle INT CHECK(Stelle BETWEEN 1 AND 5) NOT NULL,
	Pensione ENUM ('Mezza','Completa'),
	CONSTRAINT AlloggioFK_hotel FOREIGN KEY(AlloggioID) REFERENCES
Alloggio(EsercizioID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE BedAndBreakfast(
	AlloggioID INT PRIMARY KEY,
```

```
	Reception BOOLEAN,
	VietatoFumare BOOLEAN,
	ServizioBiancheria BOOLEAN,
	Lavatrice BOOLEAN,
	Lavastoviglie BOOLEAN,
	Letto ENUM ('Singolo','Matrimoniale','Singolo&Matrimoniale'),
	CONSTRAINT AlloggioFK_BB FOREIGN KEY(AlloggioID) REFERENCES
Alloggio(EsercizioID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Casa(
	AlloggioID INT PRIMARY KEY,
	Tipo ENUM ('Appartamento','Villa','Mansarda','Trullo'),
	Mq FLOAT,
	Esterno ENUM ('Giardino','Terrazzo'),
	VietatoFumare BOOLEAN,
	ServizioBiancheria BOOLEAN,
	Lavatrice BOOLEAN,
	Lavastoviglie BOOLEAN,
	Letto tipoLetto,
	CONSTRAINT AlloggioFK_casa FOREIGN KEY(AlloggioID) REFERENCES
Alloggio(EsercizioID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Attrazione(
	EsercizioID INT PRIMARY KEY,
	OrarioApertura TIME(0) NOT NULL,
	OrarioChiusura TIME(0) NOT NULL,
	PerBambini BOOLEAN,
	Tipo ENUM ('Museo','Parco','Locale'),
	CONSTRAINT EsercizioFK_attrazione FOREIGN KEY(EsercizioID)
REFERENCES Esercizio(ID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Museo(
	AttrazioneID INT PRIMARY KEY,
	Archeologico BOOLEAN,
	Artistico BOOLEAN,
	Marino BOOLEAN,
	Militare BOOLEAN,
	Scientifico BOOLEAN,
	Naturale BOOLEAN,
	Storico BOOLEAN,
	Virtuale BOOLEAN,
	CONSTRAINT AttrazioneFK_museo FOREIGN KEY(AttrazioneID)
REFERENCES Attrazione(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);

CREATE TABLE Parco(
	AttrazioneID INT PRIMARY KEY,
```

```
	Tematico BOOLEAN,
	Divertimenti BOOLEAN,
	CONSTRAINT AttrazioneFK_parco FOREIGN KEY(AttrazioneID)
REFERENCES Attrazione(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);

CREATE TABLE Locale(
	AttrazioneID INT PRIMARY KEY,
	Discoteca BOOLEAN,
	Notturno BOOLEAN,
	SalaBallo BOOLEAN,
	SalaEventi BOOLEAN,
	CONSTRAINT AttrazioneFK_locale FOREIGN KEY(AttrazioneID)
REFERENCES Attrazione(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);

CREATE TABLE PuntoRistoro(
	EsercizioID INT PRIMARY KEY,
	orarioApertura TIME(0) NOT NULL,
	orarioChiusura TIME(0) NOT NULL,
	SenzaGlutine BOOLEAN,
	VeganFriendly BOOLEAN,
	WiFi BOOLEAN,
	PostiASedere BOOLEAN,
	DaAsporto BOOLEAN,
	Servito BOOLEAN,
	Pizzeria BOOLEAN,
	Braceria BOOLEAN,
	Pub BOOLEAN,
	Ristorante BOOLEAN,
	CONSTRAINT EsercizioFK_puntoristoro FOREIGN KEY(EsercizioID)
REFERENCES Esercizio(ID) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE Pizzeria(
	PuntoRistoroID INT PRIMARY KEY,
	PizzaFritta BOOLEAN,
	SoloCena BOOLEAN,
	FornoElettrico BOOLEAN,
	FornoALegna BOOLEAN,
	CONSTRAINT PuntoRistoroFK_pizzeria FOREIGN KEY(PuntoRistoroID)
REFERENCES PuntoRistoro(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);

CREATE TABLE Braceria(
	PuntoRistoroID INT PRIMARY KEY,
	Carne BOOLEAN,
	Pesce BOOLEAN,
```

```
	CONSTRAINT PuntoRistoroFK_braceria FOREIGN KEY(PuntoRistoroID)
REFERENCES PuntoRistoro(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);

CREATE TABLE Pub(
	PuntoRistoroID INT PRIMARY KEY,
	Birreria BOOLEAN,
	CONSTRAINT PuntoRistoroFK_pub FOREIGN KEY(PuntoRistoroID)
REFERENCES PuntoRistoro(EsercizioID) ON DELETE CASCADE ON UPDATE NO
ACTION
);
```