

UNIVERSITA' DEGLI STUDI DI NAPOLI

“Federico II”

Scuola Politecnica e delle Scienze di Base

**PROGETTO
DI UNA
BASE DI
DATI**

Francesco Rosucci

Anno Accademico 2021-2022



**DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE**

VIA CLAUDIO - 80125 NAPOLI

Specifiche sui Dati

Si vuole progettare una base di dati per la gestione di una struttura che offre uffici temporanei e spazi condivisi.

I clienti ,identificati da un codice fiscale e caratterizzati da email e indirizzo (via, n.ro civico,citta,cap), possono essere privati, caratterizzati da nome, cognome ed eventuale telefono, oppure possono essere società caratterizzate da ragione sociale e p.iva. La struttura dispone di locali di tue tipi: Uffici Temporanei e spazi condivisi che possono essere sale riunioni o sale eventi. I locali sono descritti dal codice che li identifica insieme ad una capienza e al tipo di stanza. I clienti possono effettuare prenotazioni per entrambi i tipi di ambienti.Le prenotazioni sono identificate da un codice numerico e per prenotare bisogna specificare la data. Per prenotare uno spazio condiviso, inoltre, si specificano orario di inizio e fine della prenotazione; per prenotare un ufficio si specifica il numero di postazioni prenotate. Le tariffe sono relative al locale prenotato in base al tipo e alla dimensione e sono orarie in caso di spazio condiviso mentre sono giornaliere in caso di ufficio temporaneo. I clienti, nella prenotazione, possono richiedere anche dei servizi accessori(wifi, parcheggio,attrezzature quali pc, tavolette grafiche) specificandone la quantità desiderata. I servizi sono caratterizzati da una descrizione del servizio, dal prezzo e dalla disponibilità. La struttura è dotata anche di una cucina per cui i clienti possono effettuare delle ordinazioni caratterizzate da un numero di ordine, una data e un orario in cui si preferirebbe ricevere l'ordinazione e dagli alimenti scelti dal menù del giorno.

Specifiche sulle Operazioni

Per la struttura di uffici temporanei e spazi condivisi sono previste alcune operazioni, di cui riportiamo una breve descrizione ed il carico previsto.

O1 Inserisci nuovo cliente (frequenza: 10 al giorno).

O2 Modifica dati cliente (frequenza: 1 al giorno).

O3 Visualizza le prenotazioni associate ad un cliente(frequenza: 4 al giorno).

O4 Prenota un locale tenendo conto della disponibilità (frequenza: 10 al giorno).

O5 Modifica una prenotazione(data e/o ora) (frequenza: 3 al giorno).

O6 Crea una vista (frequenza: 3 al giorno).

O7 Genera una fattura per la prenotazione di un cliente (frequenza: 10 al giorno).

O8 Ricerca un cliente per nome o parte del nome (frequenza: 4/giorno).

O9 Visualizza il numero di privati e di società clienti della struttura
(frequenza: 1 al mese).

Glossario dei Termini

Di seguito una prima porzione del *Glossario Dei Termini*, risultante dall'*analisi dei requisiti*.

Termine	Descrizione	Sinonimi	Termini Collegati
Cliente	Cliente della struttura		Tipologia di cliente
Locale	Locale prenotabile da un cliente	Stanza	Ufficio temporaneo, spazio condiviso
Ufficio Temporaneo	Ufficio della struttura di cui si possono prenotare delle postazioni		
Spazio condiviso	Sala riunioni o Sala eventi prenotabile		
Servizio	Servizio aggiuntivo a pagamento		Prenotazione
Tariffa	Tariffa legata al costo di uno spazio condiviso o di una postazione di un ufficio temporaneo		Prenotazione, Locale
Fattura	Fattura rilasciata a seguito di una prenotazione		Cliente, prenotazione
Comanda	Ordinazione alla cucina	Ordine	Alimenti
Menù	Lista di alimenti che è possibile ordinare in un particolare giorno		Comanda

Progettazione Concettuale

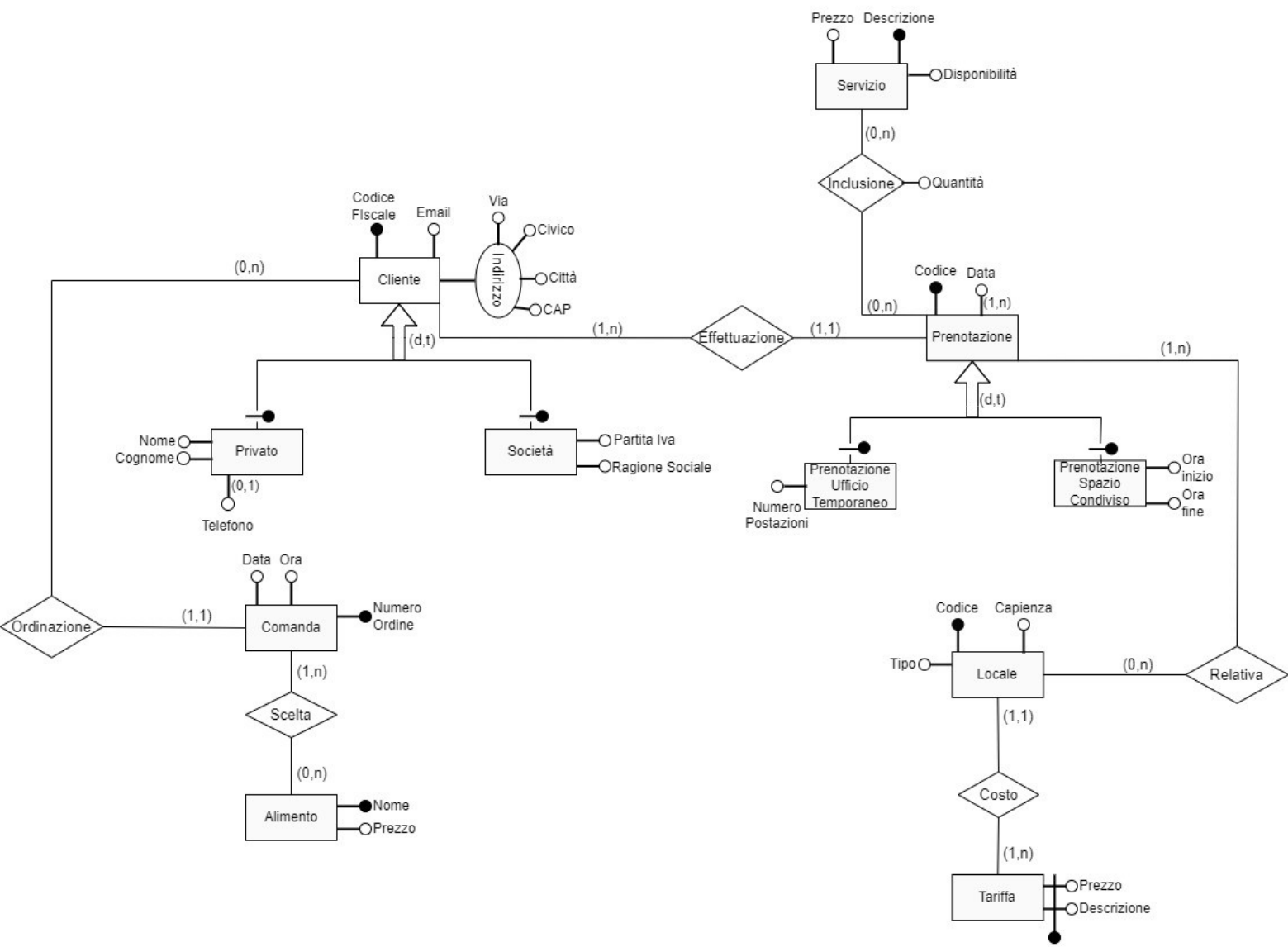
Sulla base dell'analisi dei requisiti costruisco un primo **schema scheletro** che riportiamo di seguito.



Per arrivare allo schema E-R finale continuiamo con l'analisi dei requisiti:

- 1) Il cliente è stato distinto in privato o azienda tramite una gerarchia di specializzazione totale e disgiunta;
- 2) La prenotazione è stata distinta in prenotazione ufficio temporaneo e prenotazione spazio condiviso tramite una gerarchia di specializzazione totale e disgiunta;
- 3) Sono state aggiunte le altre entità coinvolte nello schema quali: Servizio, Tariffa, Alimento con le relative associazioni;
- 4) Sono stati assegnati gli attributi alle entità coinvolte;
- 5) E' stato completato il dizionario dei dati;

Lo **schema E-R finale** è il seguente:



Entità	Descrizione	Attributi	Identificatore
Cliente	Cliente della struttura	Codice fiscale, email, indirizzo	Codice fiscale
Società	Una società cliente della struttura	Ragione sociale, Partita iva	<i>Vedi Cliente</i>
Privato	Un privato cliente della struttura	Nome, Cognome, Telefono	<i>Vedi Cliente</i>
Prenotazione	Prenotazione di uno o più Locali indicando la/le data/e in cui si intende prenotarli	Data, Codice	Codice
Prenotazione ufficio temporaneo	Prenotazione di una o più postazioni di un ufficio temporaneo	Numero postazioni	<i>Vedi Prenotazione</i>
Prenotazione spazio condiviso	Prenotazione di uno o più spazi condivisi specificando la fascia oraria in cui si vuole occupare il suddetto spazio	Ora inizio, Ora fine	<i>Vedi Prenotazione</i>
Servizio	Servizio aggiuntivo prenotabile	Prezzo, Descrizione, Disponibilità	Descrizione
Locale	Locale prenotabile della struttura	Codice, Capienza, Tipo	Codice
Tariffa	Costo giornaliero di una postazione di un ufficio o costo orario di uno spazio condiviso	Prezzo Descrizione	Prezzo, Descrizione
Comanda	Ordinazione fatta da un cliente	Data, Ora, Numero ordine	Numero ordine
Alimento	Cibo o bevanda disponibile nella giornata corrente	Nome, Prezzo	Nome

*Dizionario dei dati: **Entità***

Relazione	Entità Partecipanti	Descrizione	Attributi
Effettuazione	Cliente, Prenotazione	Un cliente effettua una prenotazione	
Relativa	Prenotazione, Locale	Associa la prenotazione con il/i locale/i prenotati	
Costo	Locale, Tariffa	E' presente una tariffa oraria o giornaliera che dipende dal locale	
Inclusione	Prenotazione,	Oltre ad un locale è	Quantità

	Servizio	possibile prenotare dei servizi aggiuntivi	
Ordinazione	Cliente, Comanda	Un cliente può effettuare un'ordinazione alla cucina della struttura	
Scelta	Comanda, Alimento	Associa la comanda con gli alimenti scelti	

Dizionario dei dati:
Relazioni

Concetto	Tipo	Volume
Cliente	E	1000
Società	E	100
Privato	E	900
Prenotazione	E	1000
Prenotazione ufficio temporaneo	E	700
Prenotazione spazio condiviso	E	300
Servizio	E	5
Locale	E	300
Tariffa	E	6
Comanda	E	400
Alimento	E	10
Effettuazione	R	1000
Inclusione	R	600
Ordinazione	R	400

Tavola Dei Volumi

Operazione	Descrizione	Frequenza	Tipo
O1	Inserimento nuovo cliente	10/giorno	OL
O2	Modifica dati di un cliente	1/giorno	OL
O3	Vedi le prenotazioni associate ad un cliente	4/giorno	OL
O4	Prenota un locale tenendo conto della disponibilità	10/giorno	OL
O5	Modifica una prenotazione (data e/o ora)	3/giorno	OL
O6	Creare una vista	3/giorno	B
O7	Genera fattura per la prenotazione di un cliente	10/giorno	OL
O8	Ricerca un cliente per nome o parte del nome	4/giorno	OL
O9	Visualizza il numero di privati e di società clienti della struttura	1/mese	B

Tavola Delle Frequenze

Progettazione Logica

Effettuiamo a questo punto la *progettazione logica* della base di dati.

Prima di procedere è necessario effettuare l'operazione di ***ristrutturazione***:

- ***Eliminazione Gerarchie***

Nello schema E-R sono presenti due gerarchie: tra l'entità padre Cliente e le sue

entità figlie Privato e Società; tra l'entità padre Prenotazione e le sue entità figlie

Prenotazione Ufficio Temporaneo e Prenotazione Spazio Condiviso.

Ristrutturo entrambe le gerarchie includendo le entità figlio nelle entità padre in

quanto noto che le entità figlie non hanno esplicite associazioni con altre entità e

non ci sono operazioni che fanno differenza tra le occorrenze di queste entità.

- ***Scelta Identificatori Primari***

Tutte le entità hanno un *unico identificatore primario*, quindi, non è necessario sceglierne uno per entità;

- ***Eliminazione di attributi multivalore/composti***

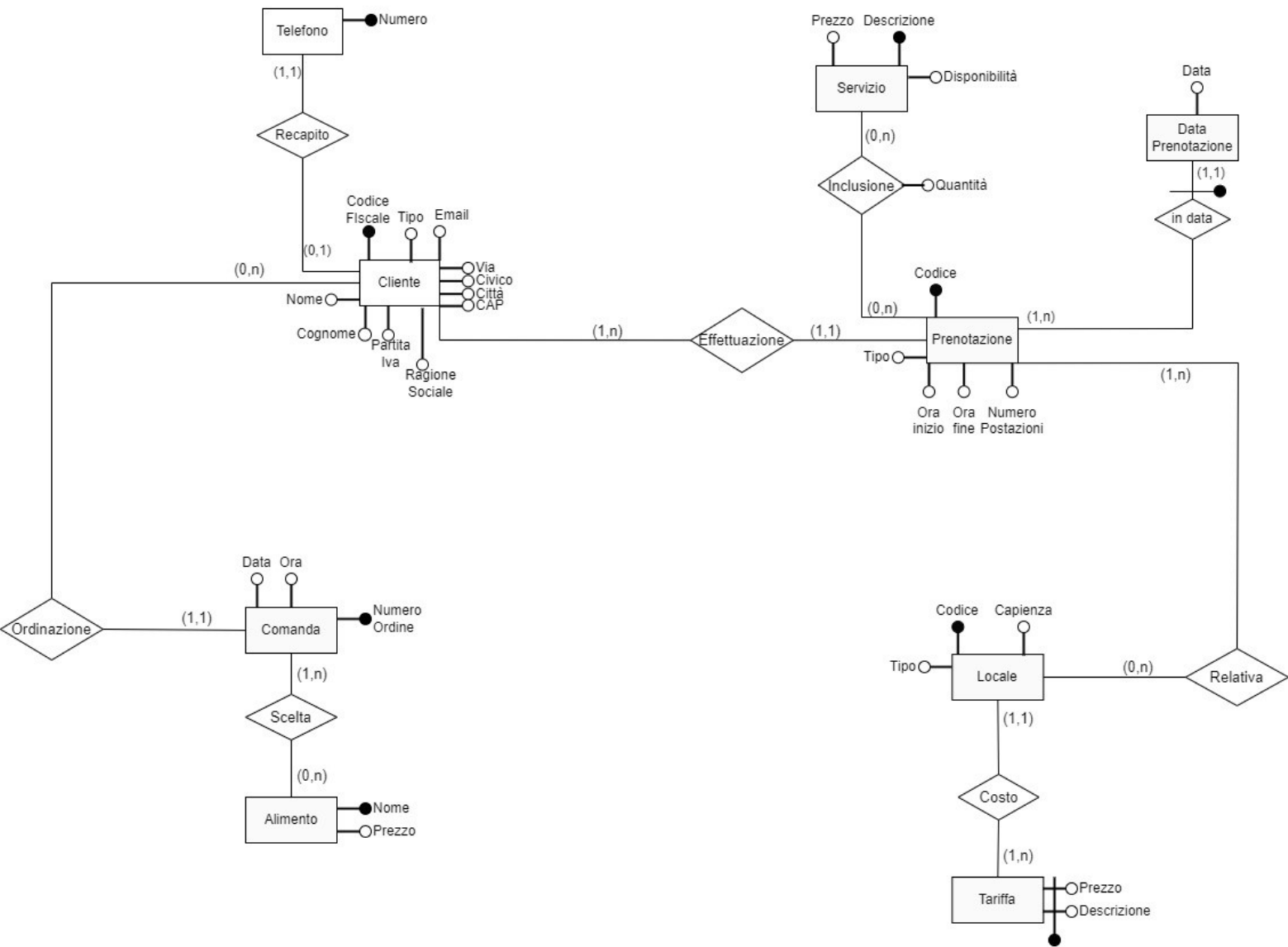
Nello schema E-R sono presenti due attributi multivalore (telefono e data) e un attributo composto (indirizzo). L'attributo composto lo ristrutturò scomponendolo in

attributi semplici mentre gli attributi multivalore li scompongo introducendo nuove

entità associate a quelle che possedevano gli attributi stessi.

- ***Partizionamento/Accorpamento Entità/Relazioni***

Non vi sono entità legate da relazioni di tipo uno a uno che possono essere accorpate per aumentare l'efficienza;



Una volta *ristrutturato* lo schema E-R, passiamo alla **traduzione** dividendola in due parti:

-Traduzione Entità

Individuiamo seguenti schemi di **R-relazione** ottenuti dalla *traduzione* delle *entità* presenti nello schema E-R ristrutturato:

Clienti (**CodiceFiscale**, Tipo, email, Via, Civico, Cap, Città, Nome, Cognome, Piva, RagioneSociale)

Telefono(**Numero**)

Prenotazioni(**Codice**, Tipo, OraInizio,OraFine,NPostazioni)

DatePrenotazioni(Data, **Codice**:PRENOTAZIONI)

Servizi(Prezzo, **Descrizione**, Disponibilità)

Locali(**Codice**, Capienza, Tipo)

Tariffe(**Prezzo**, **Descrizione**)

Comande(**Nordine**, Data, Ora)

Alimenti(**Nome**, Prezzo)

-Traduzione Associazioni

Consideriamo prima le relazioni di tipo *uno a uno*, poi quelle di tipo *uno a molti* ed infine quelle di tipo *molti a molti*.

1) *Uno a Uno*

-**Recapito** : Viene tradotta aggiungendo l'attributo Proprietario all'interno dell'entità **Telefono**;

2) *Uno a Molti*

-**Effettuazione**: Viene tradotta aggiungendo nell'entità **Prenotazioni** l'identificatore esterno Cliente:CLIENTI;

-**Indata**: Viene tradotta insieme all'entità DatePrenotazioni tramite la chiave esterna Codice;

-**Costo**: Viene tradotta aggiungendo l'identificatore esterno Prezzo:TARIFFE e

Descrizione:TARIFFE nell'entità **Locali**;

-**Ordinazione**: Viene tradotta aggiungendo l'identificatore esterno
Cliente:CLIENTI
all'entità **Comande**;

3) *Molti a Molti*

-**Inclusione**: Viene tradotta andando a creare una tabella **PrenotazioniServizi**
che
avrà **Prenotazione**:PRENOTAZIONI e **Servizio**:SERVIZI come
identificatori
primari e quantità come attributo

-**Relativa**: Viene tradotta andando a creare una tabella **PrenotazioniLocali** che
avrà
Prenotazione:PRENOTAZIONI e **Locale**:LOCALI come identificatori
primari

-**Scelta**: Viene tradotta andando a creare una tabella **AlimentiComande** che avrà
Comanda:
COMANDE e **Alimento**:ALIMENTI come identificatori primari

Lo **schema logico** finale è:

Clienti (**CodiceFiscale**, Tipo, email, Via, Civico, Cap, Città, Nome, Cognome, Piva, RagioneSociale);

Telefono(**Numero**, Proprietario:CLIENTI);

Prenotazioni(**Codice**, Tipo, OraInizio,OraFine,Npostazioni, Cliente:CLIENTI);

DatePrenotazioni(Data, **Codice**:PRENOTAZIONI);

Servizi(Prezzo, **Descrizione**, Disponibilità);

Locali(**Codice**, Capienza, Tipo, Prezzo:TARIFFE,Descrizione:TARIFFE);

Tariffe(**Prezzo**, **Descrizione**);

Comande(**Nordine**, Data, Ora, Cliente:CLIENTI);

Alimenti(**Nome**, Prezzo);

PrenotazioniServizi(**Prenotazione**:PRENOTAZIONI,

Servizio:SERVIZI,Quantità);

Implementazione delle Operazioni

Vediamo adesso l'implementazione delle operazioni che sono state definite per la struttura di uffici e spazi condivisi

- **Script di Installazione**

Di seguito la sequenza di comandi SQL da utilizzare per definire lo schema associato alla nostra applicazione, il cui nome è **SUTSC** (Struttura uffici temporanei e spazi condivisi):

```
#####
```

```
# Script di Installazione
```

```
#####
```

```
#####
```

```
create database SUTSC;
```

```
create table Clienti
```

```
(  
CodiceFiscale    char(16) NOT NULL,  
Tipo             char(8) DEFAULT 'Privato' ,  
Email           varchar(35) NOT NULL,  
Via             varchar(30) NOT NULL,  
Civico          varchar(4) NOT NULL,  
Cap             number(5) NOT NULL,  
Città           varchar(20) NOT NULL,  
Nome            varchar(20),  
Cognome         varchar(30),  
Piva            number(11) UNIQUE,  
RagioneSociale  varchar(30),  
CONSTRAINT pk_cliente PRIMARY KEY(CodiceFiscale),  
CONSTRAINT piva_constraint CHECK((tipo='Società' AND Piva is not null) or tipo='Privato' )  
);
```

```
create table Telefono
```

```
(  
Numero          number(10) NOT NULL,  
Proprietario    char(16) UNIQUE NOT NULL,  
CONSTRAINT pk_telefono PRIMARY KEY(Numero),  
CONSTRAINT fk_telefono FOREIGN KEY(Proprietario) REFERENCES CLIENTI(CodiceFiscale) ON DELETE  
CASCADE  
);
```

```
create table Prenotazioni
```

```
(  
Codice          number GENERATED BY DEFAULT ON NULL AS IDENTITY,  
Tipo            char(30) NOT NULL,  
OraInizio       number(4,2),  
OraFine         number(4,2),
```

```

Npostazioni    number(2),
Cliente        char(16) NOT NULL,
CONSTRAINT pk_prenotazioni PRIMARY KEY(Codice),
CONSTRAINT fk_prenotazioni FOREIGN KEY(Cliente) REFERENCES CLIENTI(CodiceFiscale),
CONSTRAINT booking_constraint CHECK(((tipo='Ufficio Temporaneo' AND Npostazioni is not null) or
(Orainizio is not null and Orafine is not null))
);
create table DatePrenotazioni
(
Data           date NOT NULL,
Codice         number NOT NULL,
CONSTRAINT pk_dateprenotazioni PRIMARY KEY(Codice,Data),
CONSTRAINT fk_dataprenotazioni FOREIGN KEY(Codice) REFERENCES Prenotazioni(Codice) ON
DELETE CASCADE
);

create table Servizi
(
Prezzo         number(4,2) NOT NULL,
Descrizione    varchar(40) NOT NULL,
Disponibilità  number DEFAULT 0,
CONSTRAINT pk_servizi PRIMARY KEY(Descrizione),
CONSTRAINT negative_price CHECK(Prezzo>0)
);

create table Tariffe
(
Prezzo         number(4,2) NOT NULL,
Descrizione    varchar(40) NOT NULL,
CONSTRAINT pk_tariffe PRIMARY KEY(Prezzo,Descrizione),
CONSTRAINT tariffe_price CHECK(Prezzo>0)
);

create table Locali
(
Codice         char(4) NOT NULL,
Tipo           varchar(30) NOT NULL,
Capienza       number NOT NULL,
Prezzo         number(4,2) NOT NULL,
Descrizione    varchar(40) NOT NULL,
CONSTRAINT pk_locali PRIMARY KEY(Codice),
CONSTRAINT fk_locali FOREIGN KEY(Prezzo,Descrizione) REFERENCES TARIFFE(Prezzo,Descrizione)
ON DELETE SET NULL
);

create table Comande
(
Nordine        number(4) NOT NULL,
Data           date NOT NULL,
Ora            char(5),
Cliente        char(16) NOT NULL,
CONSTRAINT pk_comande PRIMARY KEY(Nordine),
CONSTRAINT fk_comande FOREIGN KEY(Cliente) REFERENCES CLIENTI(CodiceFiscale)
);

create table Alimenti
(

```

```

Nome          varchar(20) NOT NULL,
Prezzo        number(4,2) NOT NULL,
CONSTRAINT pk_alimenti PRIMARY KEY(Nome),
CONSTRAINT prezzo_alimenti CHECK (Prezzo>0)
);

```

```

create table PrenotazioniServizi
(
Prenotazione  number NOT NULL,
Servizio      varchar(40) NOT NULL,
Quantita      number      DEFAULT 0,
CONSTRAINT pk_prenotazioni_servizi PRIMARY KEY(Prenotazione,Servizio),
CONSTRAINT fk_prenotazioni_servizi_p FOREIGN KEY(Prenotazione) REFERENCES
PRENOTAZIONI(Codice) ON DELETE CASCADE,
CONSTRAINT fk_prenotazioni_servizi_S FOREIGN KEY(Servizio) REFERENCES SERVIZI(Descrizione)
);

```

```

create table PrenotazioniLocali
(
Prenotazione  number NOT NULL,
Locale        char(4) NOT NULL,
CONSTRAINT pk_prenotazioni_locali PRIMARY KEY(Prenotazione,Locale),
CONSTRAINT fk_prenotazioni_locali_p FOREIGN KEY(Prenotazione) REFERENCES
PRENOTAZIONI(Codice) ON DELETE CASCADE,
CONSTRAINT fk_prenotazioni_locali_l FOREIGN KEY(Locale) REFERENCES LOCALI(Codice)
);

```

```

create table AlimentiComande
(
Comanda       number(4) NOT NULL,
Alimento      varchar(20) NOT NULL,
CONSTRAINT pk_alimenti_comande PRIMARY KEY(Comanda,Alimento),
CONSTRAINT fk_alimenti_comande_c FOREIGN KEY(Comanda) REFERENCES COMANDE(Nordine) ON
DELETE CASCADE,
CONSTRAINT fk_alimenti_comande_a FOREIGN KEY(Alimento) REFERENCES ALIMENTI(Nome)
);

```

Una volta create le tabelle con il comando "alter session set NLS_DATE_FORMAT = 'dd/mm/yyyy';" modifichiamo il formato della data.

- **Popolamento**

Prima di mostrare gli esempi delle possibili operazioni sul database è opportuno
popolare la base di dati:

*/*Inserimento Clienti*/*

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città, Nome,Cognome)
VALUES('RSCFNC00H15F839I','Privato','f.rosucci@studenti.unina.it', 'Luca Giordano','25',
80131,'Napoli', 'Francesco','Rosucci');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città, Nome,Cognome)
VALUES('SVRTBR89G12F839I','Privato','tiberiosaverio@gmail.com', 'Pino Mauro','13',
80128,'Napoli', 'Tiberio','Saverio');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città, Nome,Cognome)
VALUES('DRZMRK00M57X000X','Privato','marikadrazza@libero.it', 'Giuseppe Mazzini','42b',
73100,'Lecce', 'Marika','Drazza');
```

```
INSERT INTO CLIENTI(CodiceFiscale, Email, Via, Civico, Cap, Città, Nome,Cognome)
VALUES('RSIMRO90H08K123P','mriorossi@gmail.com', 'San Paolo', '8', 83040,'Avellino',
'Mario','Rossi');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città,Piva,RagioneSociale)
VALUES('ABCDEF99O23I987P','Società','fratellisrl@studiofratelli.com', 'Tizio Caio', '15',
81040,'Caserta',86334519757, 'Fratelli Srl');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città,Piva,RagioneSociale)
VALUES('FGHQID29O12U647H','Società', 'lampade&lampade@gmail.com', 'Giuseppe
Garibaldi', '121c', 80142,'Napoli',56915478632, 'Lampade&Lampade');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città,Piva,RagioneSociale)
VALUES('PLSHYT44L09Q456N','Società', 'esposito@espositolegge.com', 'Giovanni Falcone', '49',
84123,'Salerno',86334517757, 'Esposito Legge');
```

*/*Inserimento Telefono*/*

```
INSERT INTO TELEFONO VALUES(3482310923, 'RSCFNC00H15F839I');
```

```
INSERT INTO TELEFONO VALUES(3809821532, 'RSIMRO90H08K123P');
```

```
INSERT INTO TELEFONO VALUES(3339750412, 'DRZMRK00M57X000X');
```

*/*Inserimento Prenotazioni*/*

```
INSERT INTO PRENOTAZIONI(Tipo,Npostazioni,Cliente) VALUES('Ufficio Temporaneo', 3,
'RSCFNC00H15F839I');
```

```
INSERT INTO PRENOTAZIONI(Tipo,Npostazioni,Cliente) VALUES('Ufficio Temporaneo', 1,
'SVRTBR89G12F839I');
```


INSERT INTO PRENOTAZIONI(Tipo,Oralnizio,OraFine,Cliente) VALUES('Sala Riunioni', 10.00, 12.00, 'DRZMRK00M57X000X');

INSERT INTO PRENOTAZIONI(Tipo,Npostazioni,Cliente) VALUES('Ufficio Temporaneo', 1, 'RSIMRO90H08K123P');

INSERT INTO PRENOTAZIONI(Tipo,Oralnizio,OraFine,Cliente) VALUES('Sala eventi', 14.00,18.00, 'ABCDEF99O23I987P');

INSERT INTO PRENOTAZIONI(Tipo,Oralnizio,OraFine,Cliente) VALUES('Sala riunioni', 15.00,16.30, 'FGHQID29O12U647H');

INSERT INTO PRENOTAZIONI(Tipo, Npostazioni,Cliente) VALUES('Ufficio Temporaneo', 10, 'PLSHYT44L09Q456N');

/*Inserimento DatePrenotazioni*/

INSERT INTO DATEPRENOTAZIONI VALUES('20-02-2022', 1);

INSERT INTO DATEPRENOTAZIONI VALUES('21-02-2022', 1);

INSERT INTO DATEPRENOTAZIONI VALUES('22-02-2022', 1);

INSERT INTO DATEPRENOTAZIONI VALUES('16-03-2022', 2);

INSERT INTO DATEPRENOTAZIONI VALUES('17-03-2022', 2);

INSERT INTO DATEPRENOTAZIONI VALUES('18-03-2022', 2);

INSERT INTO DATEPRENOTAZIONI VALUES('19-03-2022', 2);

INSERT INTO DATEPRENOTAZIONI VALUES('16-03-2022',3);

INSERT INTO DATEPRENOTAZIONI VALUES('20-02-2022', 4);

INSERT INTO DATEPRENOTAZIONI VALUES('21-02-2022', 4);

INSERT INTO DATEPRENOTAZIONI VALUES('22-02-2022', 4);

INSERT INTO DATEPRENOTAZIONI VALUES('14-06-2022',5);

INSERT INTO DATEPRENOTAZIONI VALUES('14-06-2022',6);

INSERT INTO DATEPRENOTAZIONI VALUES('12-03-2022',7);

INSERT INTO DATEPRENOTAZIONI VALUES('13-03-2022',7);

INSERT INTO DATEPRENOTAZIONI VALUES('14-03-2022',7);

/*Inserimento Tariffe*/

INSERT INTO TARIFFE VALUES(20.00,'Ufficio temporaneo postazione piccola');

INSERT INTO TARIFFE VALUES(30.00, 'Ufficio temporaneo postazione grande');

INSERT INTO TARIFFE VALUES(15.00,'Spazi condivisi piccola dimensione');

INSERT INTO TARIFFE VALUES(30.00,'Spazi condivisi media dimensione');

INSERT INTO TARIFFE VALUES(50.00, 'Spazi condivisi grandi dimensioni');

/*Inserimento Servizi*/

INSERT INTO SERVIZI VALUES(10.50,'Wi-Fi giornaliero', 10000);

INSERT INTO SERVIZI VALUES(5, 'Parcheggio giornaliero', 200);

INSERT INTO SERVIZI VALUES(7.99, 'Tavoletta Grafica', 100);

/*Inserimento Comande*/

INSERT INTO COMANDE VALUES(1,'20-02-2022','13:00', 'RSCFNC00H15F839I');

INSERT INTO COMANDE VALUES(2,'18-03-2022', '14:30', 'SVRTBR89G12F839I');

/*Inserimento Alimenti*/

INSERT INTO ALIMENTI VALUES('Pasta e patate', 8);

INSERT INTO ALIMENTI VALUES('Pasta al sugo', 4);

INSERT INTO ALIMENTI VALUES('Arista di maiale', 6);

INSERT INTO ALIMENTI VALUES('Cosce di pollo', 5);

INSERT INTO ALIMENTI VALUES('Acqua 1L', 1.5);

INSERT INTO ALIMENTI VALUES('Acqua 0.5L', 0.50);

INSERT INTO ALIMENTI VALUES('CocaCola 33cl', 2);

/*Inserimento Locali*/

INSERT INTO LOCALI VALUES('A001','Ufficio Temporaneo', 5, 20.00,'Ufficio temporaneo postazione piccola');

INSERT INTO LOCALI VALUES('A002','Ufficio Temporaneo',4, 20.00,'Ufficio temporaneo postazione piccola');

INSERT INTO LOCALI VALUES('A050', 'Ufficio Temporaneo', 15, 30.00,'Ufficio temporaneo postazione grande');

INSERT INTO LOCALI VALUES('B001', 'Sala Riunioni', 10, 15.00,'Spazi condivisi piccola dimensione');

INSERT INTO LOCALI VALUES('C003','Sala eventi',50, 30.00,'Spazi condivisi media dimensione');

/*Inserimento AlimentiComande*/

INSERT INTO ALIMENTICOMANDE VALUES(1,'Pasta al sugo');

INSERT INTO ALIMENTICOMANDE VALUES(1,'Arista di maiale');

INSERT INTO ALIMENTICOMANDE VALUES(1,'Acqua 1L');

INSERT INTO ALIMENTICOMANDE VALUES(2,'Coscette di pollo');

INSERT INTO ALIMENTICOMANDE VALUES(2,'CocaCola 33cl');

/*Inserimento PrenotazioniServizi*/

INSERT INTO PRENOTAZIONISERVIZI VALUES(1, 'Parcheggio giornaliero',3);

INSERT INTO PRENOTAZIONISERVIZI VALUES(1, 'Wi-Fi giornaliero',3);

INSERT INTO PRENOTAZIONISERVIZI VALUES(3, 'Tavoletta Grafica',1);

/*Inserimento PrenotazioniLocali*/

INSERT INTO PRENOTAZIONILOCALI VALUES(1,'A001');

INSERT INTO PRENOTAZIONILOCALI VALUES(2,'A002');

INSERT INTO PRENOTAZIONILOCALI VALUES(3,'B001');

INSERT INTO PRENOTAZIONILOCALI VALUES(4,'A001');

INSERT INTO PRENOTAZIONILOCALI VALUES(5,'C003');

INSERT INTO PRENOTAZIONILOCALI VALUES(6,'B001');

INSERT INTO PRENOTAZIONILOCALI VALUES(7,'A050');

• OP_1: “Inserimento nuovo cliente”

Si effettua una INSERT con i dati forniti dal cliente. Di seguito i casi sia per clienti privati che per società

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap, Città,
Nome,Cognome)
VALUES('MRTVRN69C49F839T','Privato','f.rosucci@studenti.unina.it', 'Mario
Fiore','21', 80131,'Napoli', 'Veronica','Morato');
```

```
INSERT INTO CLIENTI(CodiceFiscale,Tipo, Email, Via, Civico, Cap,
Città,Piva,RagioneSociale)
VALUES('CCCCMP072D09K222D','Società','limpoco@gmail.com', 'Dante
```

• OP_2: “Modifica dati di un cliente”

Supponiamo che un cliente abbia cambiato email, per cambiare questa informazione sul nostro

database ricorreremo all'istruzione UPDATE.

Prima dell'update:

CODICEFISCALE	TIPO	EMAIL	VIA	CIVICO	CAP	CITTÀ	NOME	COGNOME
RSCFNC00H15F839I	Privato	f.rosucci@studenti.unina.it	Luca Giordano	25	80131	Napoli	Francesco	Rosucci

Dopo l'update:

CODICEFISCALE	TIPO	EMAIL	VIA	CIVICO	CAP	CITTÀ	NOME	COGNOME
RSCFNC00H15F839I	Privato	francescorosucci@libero.it	Luca Giordano	25	80131	Napoli	Francesco	Rosucci

Se volessimo aggiornare il numero di telefono potremmo fare come segue

```
UPDATE TELEFONO SET  
Numero=3332315437
```

Verificando il successo dell'update tramite:

```
SELECT Nome,Cognome,Numero  
FROM telefono JOIN clienti ON telefono.proprietario=clienti.codicefiscale
```

- **OP_3: “Vedi le prenotazioni associate ad un cliente”**

Per vedere tutte le prenotazioni effettuate da un cliente utilizziamo l'istruzione SELECT:

```
SELECT *  
FROM prenotazioni  
WHERE cliente = 'RSCFNC00H15F839I';
```

CODICE	TIPO	ORAINIZIO	ORAFINE	NPOSTAZIONI	CLIENTE
1	Ufficio Temporaneo	-	-	3	RSCFNC00H15F839I

- **OP_4: “Prenota un locale tenendo conto della disponibilità”**

Per effettuare una prenotazione dobbiamo prima verificare che ci siano locali disponibili nella data o nell'ora in cui vogliamo prenotare. Supponiamo di voler prenotare una postazione di un ufficio il 16 marzo 2022. Dovremmo effettuare un controllo e se ci sono postazioni libere inserire la prenotazione.

Per verificare quali locali sono disponibili in questa data utilizziamo la seguente query innestata:

```
SELECT *  
FROM Locali  
WHERE data NOT IN (SELECT data FROM prenotazioni  
WHERE proprietario = 'RSCFNC00H15F839I');
```

Ottenendo quanto segue (coerente poichè il locale A002 e B001 sono prenotati in suddetta data):

CODICE	TIPO	CAPENZA	PREZZO	DESCRIZIONE
A001	Ufficio Temporaneo	5	20	Ufficio temporaneo postazione piccola
A050	Ufficio Temporaneo	15	30	Ufficio temporaneo postazione grande
C003	Sala eventi	50	30	Spazi condivisi media dimensione

Se volessimo visualizzare anche un elenco di quali uffici temporanei sono prenotati ma non eccedono la loro capienza potremmo usare i seguenti comandi sfruttando PL/SQL (Per l'esempio si è supposto di voler vedere i posti liberi dei locali prenotati il 20 febbraio 2022)

```

DECLARE
postiliberi prenotazioni.npostazioni%TYPE;
capienza locali.capienza%TYPE;
counter NUMBER:=0;
postioccupati prenotazioni.npostazioni%TYPE;
locale_var locali.codice%TYPE;
type array_t is varray(10) of varchar(10);
arraylocali array_t:= array_t();

BEGIN
    FOR pren IN (SELECT distinct locale FROM prenotazioni pr join dateprenotazioni dpr on
pr.codice=dpr.codice join prenotazionilocali plo on pr.codice=plo.prenotazione join locali lo on
plo.locale=lo.codice
    where pr.tipo='Ufficio Temporaneo' AND data='20/02/2022')
        LOOP
            counter := counter + 1;
            arraylocali.extend();
            arraylocali(counter) := pren.locale;
        END LOOP;

    for i in 1..arraylocali.count loop

        select distinct capienza
        into capienza
        from prenotazioni pr join dateprenotazioni dpr on pr.codice=dpr.codice join
prenotazionilocali plo on pr.codice=plo.prenotazione join locali lo on plo.locale=lo.codice
        where pr.tipo='Ufficio Temporaneo' AND data='20/02/2022' AND locale=arraylocali(i);

        locale_var := arraylocali(i);

        select sum(NPOSTAZIONI)
        into postioccupati
        from prenotazioni pr join dateprenotazioni dpr on pr.codice=dpr.codice join
prenotazionilocali plo on pr.codice=plo.prenotazione join  locali lo on plo.locale=lo.codice
        where pr.tipo='Ufficio Temporaneo' AND data='20/02/2022' AND locale=locale_var;

        postiliberi:=capienza-postioccupati;
        IF postiliberi>0
        THEN
            DBMS_OUTPUT.put_line('Locale:'||locale_var||' Posti Occupati:'||postioccupati||'
Capienza:'||capienza||' ->Posti liberi:'||postiliberi);
        END IF;
    
```

Ottenendo nel nostro caso (Coerente con il nostro caso poichè A001 è l'unico locale prenotato in questa data ed è prenotato da RSCFNC00H15F839I per 3 posti e da RSIMRO90H08K123P per 1 posto per un totale di 4 posti occupati su 5)

Statement processed.

Locale:A001 Posti Occupati:4 Capienza:5 ->Posti liberi:1

Si può proseguire quindi con delle insert per registrare la prenotazione

```
INSERT INTO PRENOTAZIONI(Tipo,Npostazioni,Cliente) VALUES('Ufficio Temporaneo',
1, ' MRTVRN69C49F839T ');
INSERT INTO DATEPRENOTAZIONI VALUES('16-03-2022', 8);
INSERT INTO PRENOTAZIONILOCALI VALUES(8,'A003');
```

• OP_5: “Modifica una prenotazione (data e/o ora)

Se volessimo modificare ad esempio una data della prenotazione 3 avremmo

```
UPDATE DatePrenotazioni SET Data= '11-02-2022 ' WHERE
data='16/03/2022' AND codice = 3:
```

Prima dell'update

DATA	CODICE
20/02/2022	1
21/02/2022	1
22/02/2022	1
16/03/2022	2
17/03/2022	2
18/03/2022	2
19/03/2022	2
16/03/2022	3

----->

Dopo l'update

DATA	CODICE
20/02/2022	1
21/02/2022	1
22/02/2022	1
16/03/2022	2
17/03/2022	2
18/03/2022	2
19/03/2022	2
11/02/2022	3

----->

• OP_6: “Creare una vista”

Potremmo poter creare una vista con tutti i dati su una prenotazione compreso data e locale prenotato

```
CREATE VIEW PrenotazioniIntegrali
AS
SELECT pr.codice,pr.tipo,orainizio,orafine,Npostazioni,pr.cliente,dpr.data,plo.locale
FROM Prenotazioni pr join DatePrenotazioni dpr on pr.codice = dpr.codice join
PrenotazioniLocali plo on plo.Prenotazione = pr.codice;
```

CODICE	TIPO	ORAINIZIO	ORAFINE	NPOSTAZIONI	CLIENTE	DATA	LOCALE
1	Ufficio Temporaneo	-	-	3	RSCFNC00H15F839I	20/02/2022	A001
1	Ufficio Temporaneo	-	-	3	RSCFNC00H15F839I	21/02/2022	A001
1	Ufficio Temporaneo	-	-	3	RSCFNC00H15F839I	22/02/2022	A001
2	Ufficio Temporaneo	-	-	1	SVRTBR89G12F839I	16/03/2022	A002
2	Ufficio Temporaneo	-	-	1	SVRTBR89G12F839I	17/03/2022	A002
2	Ufficio Temporaneo	-	-	1	SVRTBR89G12F839I	18/03/2022	A002
2	Ufficio Temporaneo	-	-	1	SVRTBR89G12F839I	19/03/2022	A002
3	Sala Riunioni	10:00	12:00	-	DRZMRK00M57X000X	16/03/2022	B001
4	Ufficio Temporaneo	-	-	1	RSIMRO90H08K123P	20/02/2022	A001
4	Ufficio Temporaneo	-	-	1	RSIMRO90H08K123P	21/02/2022	A001
4	Ufficio Temporaneo	-	-	1	RSIMRO90H08K123P	22/02/2022	A001
5	Sala eventi	14:00	18:00	-	ABCDEF99023I987P	14/06/2022	C003
6	Sala riunioni	15:00	16:30	-	FGHQID29012U647H	14/06/2022	B001
7	Ufficio Temporaneo	-	-	10	PLSHYT44L09Q456N	12/03/2022	A050
7	Ufficio Temporaneo	-	-	10	PLSHYT44L09Q456N	13/03/2022	A050
7	Ufficio Temporaneo	-	-	10	PLSHYT44L09Q456N	14/03/2022	A050

• OP_7: “Genera fattura per la prenotazione di un cliente”

Per generare una fattura dobbiamo moltiplicare il prezzo dell’ufficio per i giorni di prenotazione oppure Moltiplicare il prezzo dello spazio condiviso per le ore in cui è stato prenotato.

Nel caso volessimo visualizzare la fattura della prenotazione con codice 1, che si riferisce ad un ufficio temporaneo avremmo:

```
DECLARE
prezzo_giornaliero number;
giorni number;
totale number;
```



```

cliente prenotazioni.cliente%TYPE;
locale_var locali.codice%TYPE;
BEGIN
SELECT lo.prezzo
INTO prezzo_giornaliero
FROM prenotazionilocali plo join locali lo on plo.locale=lo.codice
WHERE plo.prenotazione=1;
--dbms_output.put_line(prezzo_giornaliero);
SELECT count(*)
INTO giorni
FROM dateprenotazioni
WHERE codice=1;
--dbms_output.put_line(giorni);
SELECT cliente
INTO cliente
FROM prenotazioni
WHERE codice=1;
--dbms_output.put_line(cliente);
SELECT locale
INTO locale_var
FROM prenotazionilocali
WHERE prenotazione=1;
totale:=prezzo_giornaliero*giorni;
dbms_output.put_line('FATTURA');
dbms_output.put_line('Cliente:'||cliente||' | Locale:'||locale_var||' | Prezzo giornaliero:'||prezzo_giornaliero||'€/g | Giorni di
prenotazione:'||giorni||' | Totale:'||totale||'€');
END;

```

Statement processed.

FATTURA

Cliente:RSCFNC00H15F839I | Locale:A001 | Prezzo giornaliero:20€/g | Giorni di prenotazione:3 | Totale:60€

Se volessimo la fattura della prenotazione 3 che si riferisce ad uno spazio condiviso avremmo:

```

DECLARE
prezzo_orario tariffe.prezzo%TYPE;
oraInizio prenotazioni.oraInizio%TYPE;
oraFine prenotazioni.orafine%TYPE;
totale number;
cliente prenotazioni.cliente%TYPE;
locale_var locali.codice%TYPE;
BEGIN

SELECT lo.prezzo
INTO prezzo_orario
FROM prenotazionilocali plo join locali lo on plo.locale=lo.codice
WHERE plo.prenotazione=3;

```

```
--dbms_output.put_line(prezzo_orario);
```

```
SELECT orainizio  
INTO oralnizio  
FROM prenotazioni  
WHERE codice=3;
```

```
--dbms_output.put_line(oralnizio);
```

```
SELECT orafine  
INTO oraFine  
FROM prenotazioni  
WHERE codice=3;
```

```
--dbms_output.put_line(oraFine);
```

```
SELECT cliente  
INTO cliente  
FROM prenotazioni  
WHERE codice=3;
```

```
--dbms_output.put_line(cliente);
```

```
SELECT locale  
INTO locale_var  
FROM prenotazionilocali  
WHERE prenotazione=3;
```

```
totale:=prezzo_orario*(oraFine-oralnizio);
```

```
dbms_output.put_line('FATTURA');  
dbms_output.put_line('Cliente:'||cliente||' | Locale:'||locale_var||' | Prezzo orario:'||prezzo_orario||'€/g | Ore di prenotazione:'||  
(oraFine-oralnizio)||' | Totale:'||totale||'€');
```

```
END;
```

FATTURA

Cliente:DRZMRK00M57X000X | Locale:B001 | Prezzo orario:15€/g | Ore di prenotazione:2 | Totale:30€

- **OP_8: “Ricerca un cliente per nome o parte del nome”**

L’operazione ha come obiettivo l’individuare uno o più clienti tramite il nome o parte di esso

Facciamo due esempi:

1) Tutti i clienti che si chiamano Francesco

Select

CODICEFISCALE	TIPO	EMAIL	VIA	CIVICO	CAP	CITTÀ	NOME	COGNOME
RSCFNC00H15F839I	Privato	f.rosucci@studenti.unina.it	Luca Giordano	25	80131	Napoli	Francesco	Rosucci

2) Tutte le società che hanno una a nel nome come terza lettera

```
Select CodiceFiscale,Tipo,Email,Via,Civico,Cap,Città,Piva,RagioneSociale FROM
Clienti
```

CODICEFISCALE	TIPO	EMAIL	VIA	CIVICO	CAP	CITTÀ	PIVA	RAGIONESOCIALE
ABCDEF99023I987P	Società	fratellisrl@studiofratelli.com	Tizio Caio	15	81040	Caserta	86334519757	Fratelli Srl

• **OP_9: “Visualizza il numero di privati e di società clienti della struttura”**

A questo scopo possiamo utilizzare una funzione aggregata e la clausola “group by”:

```
SELECT tipo,count(*) as NumeroClienti
FROM Clienti
GROUP BY tipo;
```

TIPO	NUMEROCLIENTI
Privato	4
Società	3