

Progetto Base di dati

Gestione centro di parrucchieria

Scozzari Vincenzo

Università degli studi di Catania – Dipartimento di Matematica e informatica

1. Caratteristiche	pag. 2
2. Analisi dei requisiti	pag. 3
• 2.1 Glossario dei termini	pag. 3
• 2.2 Analisi dei dati	pag. 4
• 2.3 Operazioni	pag. 5
3. Progettazione Concettuale: Modello E-R	pag. 6
• 3.1 Schema scheletro	pag. 6
• 3.1 Schema intermedio 1	pag. 7
• 3.1 Schema intermedio 2	pag. 8
• 3.2 Schema finale	pag. 9
• 3.3 Dizionario dei dati – Entità	pag. 10
• 3.4 Dizionario dei dati – Relazioni	pag. 11
• 3.5 Vincoli non esprimibili dallo schema E-R	pag. 11
• 3.6 Tabella volumi	pag. 12
• 3.7 Tabella frequenze	pag. 12
• 3.8 Tabella operazioni	pag. 12
• 3.9 Traduzione nel modello relazionale	pag. 13
4. Implementazione SQL	pag. 14
• 4.1 Tabelle	pag. 14
• 4.2 Operazioni	pag. 17
• 4.3 Diagramma EER	pag. 21

1. Caratteristiche

Si vuole progettare un database per la gestione di un centro di parrucchieria.

Il centro di parrucchieria offre una serie di trattamenti di bellezza mirati alla cura dei capelli.

Di ciascun trattamento si conoscono: il codice identificativo, il nome, il prezzo e la descrizione. Fra i trattamenti offerti si possono distinguere due principali categorie: uomo, donna. Il prezzo di ciascun trattamento varia in base alla categoria.

I clienti del centro di parrucchieria sono caratterizzati da un codice identificativo, dal nome, dal cognome e dal numero di telefono, possono prenotare il trattamento specificando la data e l'ora dell'appuntamento. Ciascun cliente può prenotare più trattamenti in momenti diversi.

Nel centro di parrucchieria lavorano diversi dipendenti di cui si conosce il codice fiscale, il nome, il cognome, il numero di telefono e lo stipendio. Un dipendente può eseguire sia trattamenti uomo che trattamenti donna. Un trattamento viene eseguito da un dipendente.

Per i prodotti utilizzati nei trattamenti si conoscono il codice identificativo, il nome, il tipo e il costo del lotto all'ingrosso. I prodotti saranno riforniti una volta al mese.

2. Analisi dei requisiti

2.1 Glossario dei termini

Termine	Descrizione	Sinonimi	Termini Collegati
Trattamento	Servizio offerto da parte del centro di parrucchieria	servizio	Prenotazione, dipendente, prodotto
Cliente	Colui che prenota un trattamento offerto dal centro di parrucchieria		Prenotazione
Dipendente	Colui che lavora per il centro di parrucchieria	impiegato	Trattamento
Prodotto	Prodotto utilizzato per i trattamenti nel centro di parrucchieria		Trattamento
Prenotazione	Tiene traccia dell'appuntamento preso dal cliente in merito ad un trattamento fornito dal centro di parrucchieria	appuntamento	Cliente, trattamento

2.2 Analisi dei dati

Generale:

Si vuole progettare un database per un centro di parrucchieria.

Dati sui trattamenti:

Di ciascun trattamento si conoscono: il codice identificativo, il nome, il prezzo e la descrizione. Fra i trattamenti offerti, si possono distinguere due principali categorie: uomo, donna. Il prezzo di ciascun trattamento varia in base alla categoria.

Dati sui clienti:

I clienti del centro di parrucchieria sono caratterizzati da un codice identificativo, dal nome, dal cognome e dal numero di telefono, possono prenotare il trattamento specificando la data e l'ora dell'appuntamento. Ciascun cliente può prenotare più trattamenti in momenti diversi.

Dati sulla prenotazione

Di ciascuna prenotazione si conoscono: il codice identificativo, la data, l'ora.

Dati sui dipendenti:

Nel centro di parrucchieria lavorano diversi dipendenti, di ciascun dipendente si conosce il codice fiscale, il nome, il cognome, il numero di telefono e lo stipendio.

Dati sui prodotti:

Di ciascun prodotto utilizzato nei trattamenti si conosce il codice identificativo, il nome, il tipo e il costo che il centro di parrucchieria paga al fornitore. I prodotti saranno riforniti una volta al mese.

2.3 Operazioni

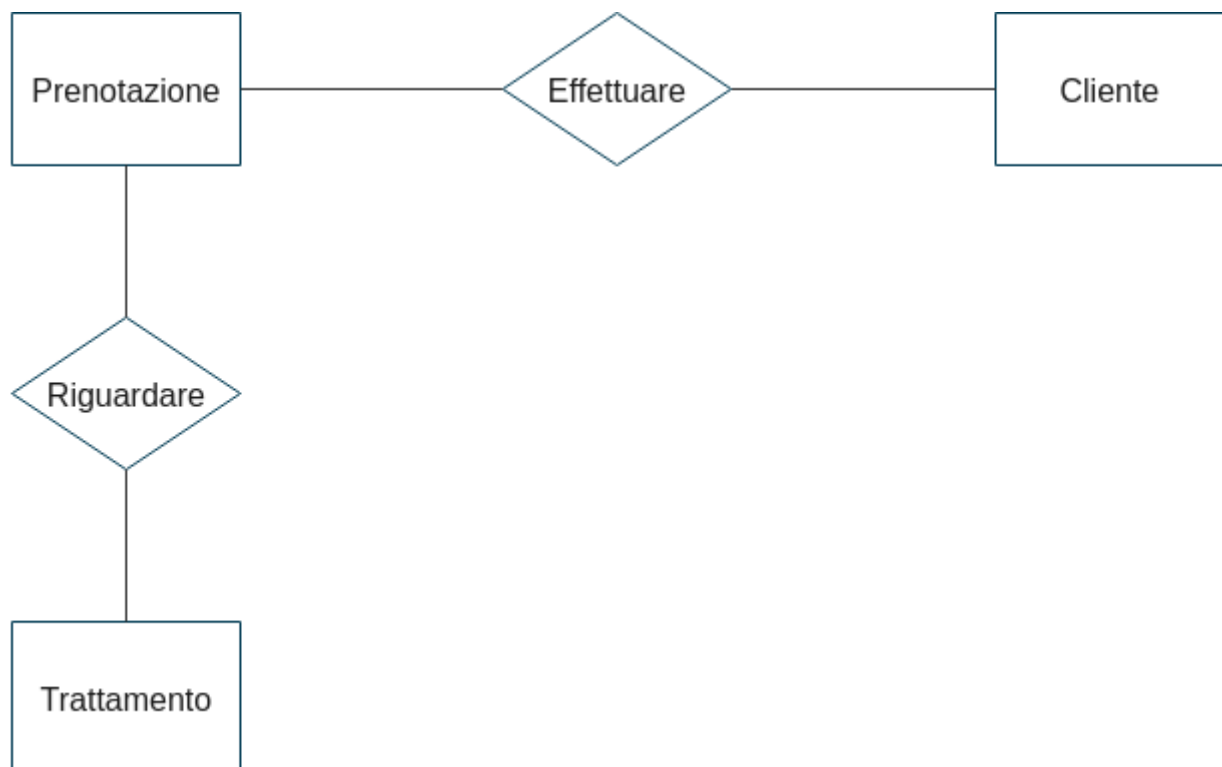
- **OP.1** Inserimento di un nuovo cliente
- **OP.2** Aggiornare il recapito telefonico di un cliente
- **OP.3** Effettuare una prenotazione in base alla disponibilità
- **OP.4** Contare il numero di prenotazioni effettuate in una data
- **OP.5** Contare quante prenotazioni ha effettuato un cliente
- **OP.6** Calcolare le entrate economiche mensili (guadagni dei trattamenti)
- **OP.7** Calcolare le uscite mensili dovute ai rifornimenti e agli stipendi
- **OP.8** Calcolare il guadagno mensile
- **OP.9** Contare quanti prodotti vengono utilizzati in un determinato trattamento

3. Diagramma E-R

Per la progettazione del diagramma verrà utilizzato un approccio di tipo TOP-DOWN

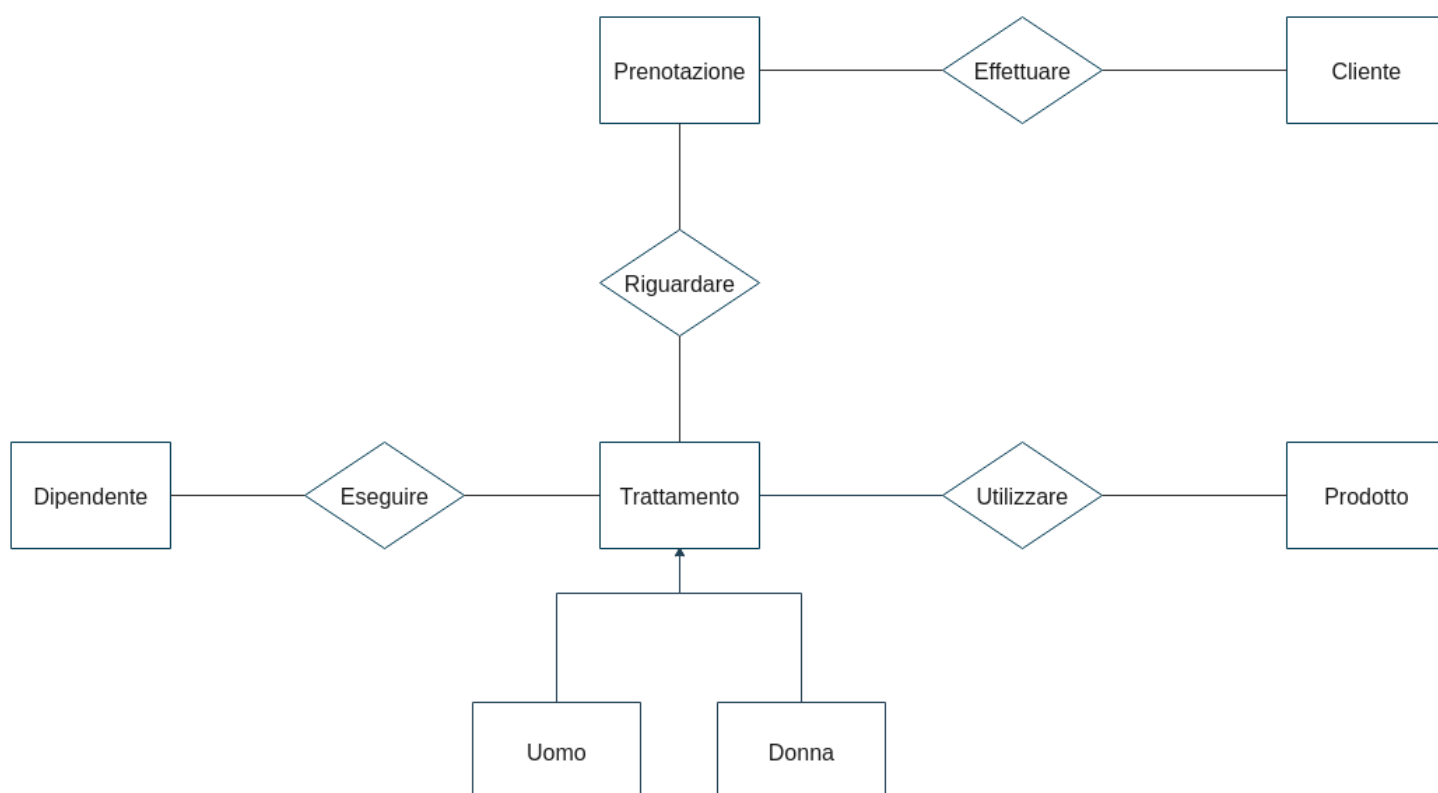
3.1 Schema iniziale

Rappresenta un'idea generale e minima dell'intera base di dati



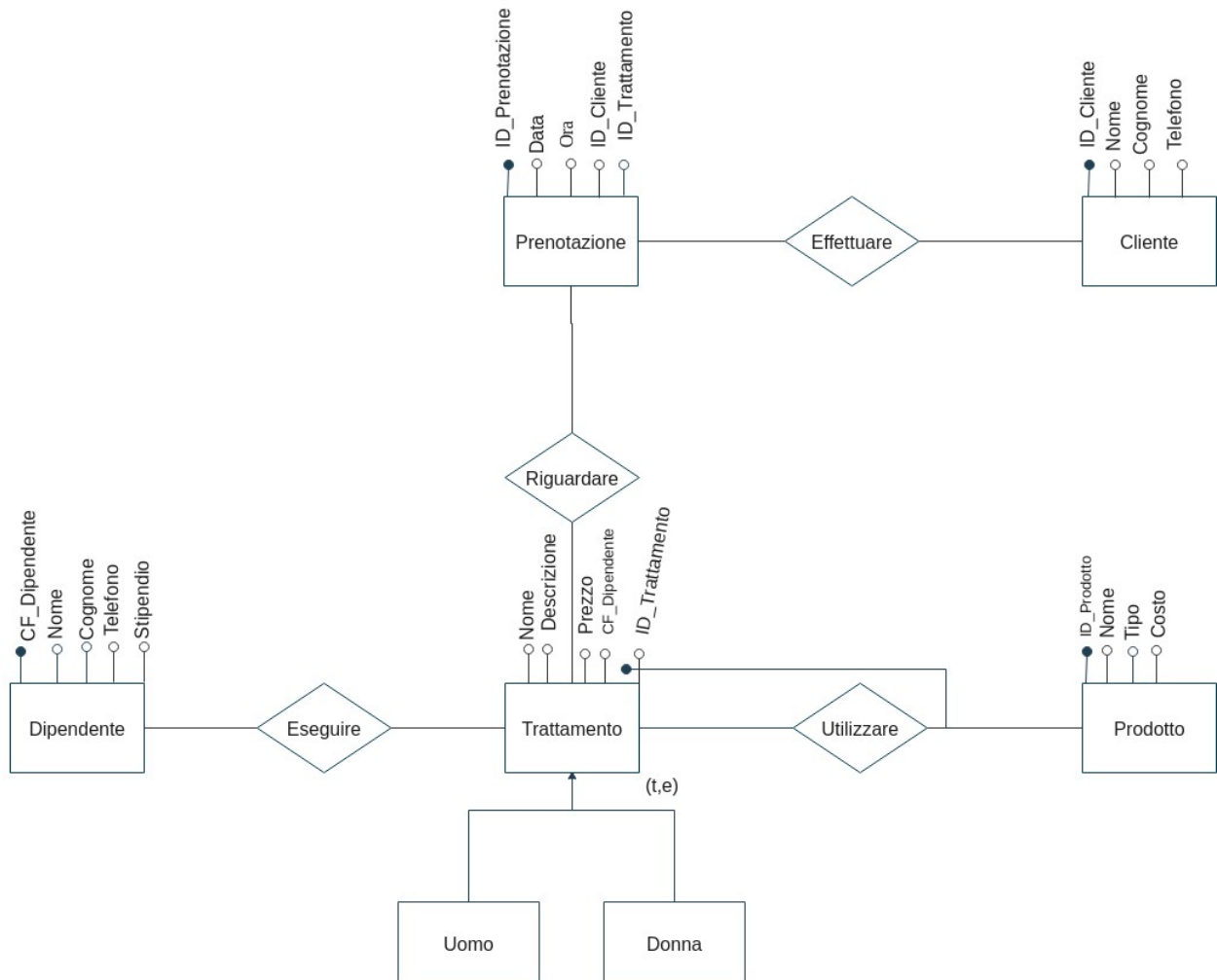
3.1 Schema intermedio 1

Vengono definite più entità e relazioni



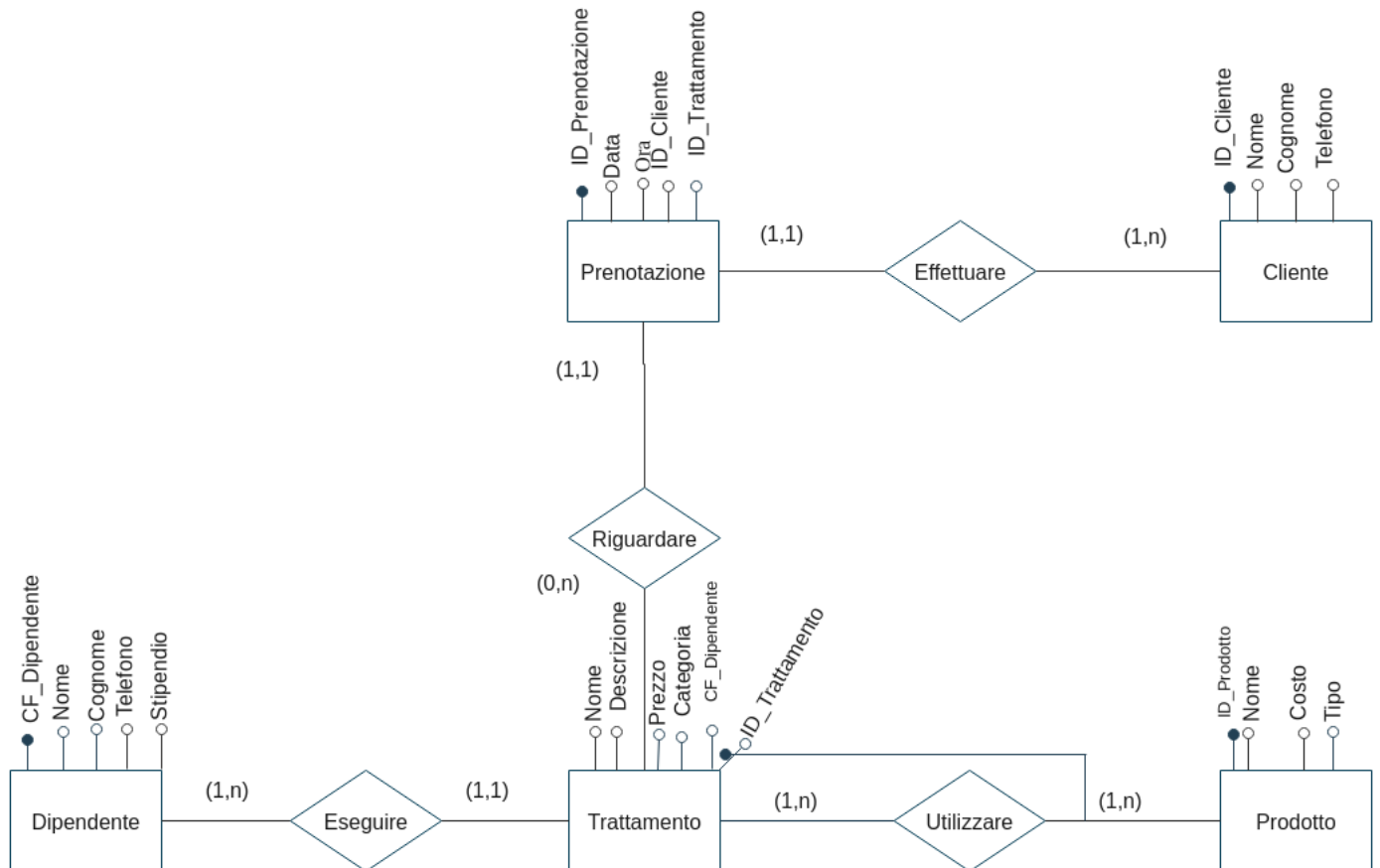
3.1 Schema Intermedio 2

Si inseriscono i dettagli di ogni elemento



3.2 Schema finale

- Si inseriscono le cardinalità
- Si elimina la generalizzazione sull'entità trattamento che è totale ed esclusiva attraverso un collasso verso l'alto.



3.3 Entità

ENTITÀ	ATTRIBUTI	DESCRIZIONE	PRIMARY_KEY
CLIENTE	ID_Cliente, Nome, Cognome, Telefono	Persona che prenota un trattamento	ID_Cliente
PRENOTAZIONE	ID_Prenotazione, Data, Ora, ID_Cliente, ID_Trattamento	Serve per stabilire un appuntamento in merito ad un trattamento	ID_Prenotazione
TRATTAMENTO	ID_Trattamento, ID_Prodotto, Nome, Descrizione, Prezzo, Categoria, CF_Dipendente	Servizio offerto dal centro di parrucchieria	ID_Trattamento, ID_Prodotto
DIPENDENTE	CF_Dipendente, Nome, Cognome, Telefono , Stipendio	Persone che collaborano all'interno del centro di parrucchieria	CF_Dipendente
PRODOTTO	ID_Prodotto, Nome, Tipo, Costo	Prodotto utilizzato dal centro di parrucchieria	ID_Prodotto

3.4 Relazioni

RELAZIONE	ENTITÀ	DESCRIZIONE
EFFETTUARE	CLIENTE, PRENOTAZIONE	Cliente effettua una prenotazione
RIGUARDARE	PRENOTAZIONE, TRATTAMENTO	La prenotazione richiede un trattamento
ESEGUIRE	DIPENDENTE, TRATTAMENTO	Il dipendente esegue un trattamento
UTILIZZARE	TRATTAMENTO, PRODOTTO	Il trattamento utilizza un prodotto

3.5 Vincoli

- Possono essere accettate al più 3 prenotazioni aventi la stessa data e la stessa ora.
- I codici fiscali saranno di lunghezza massima di 16 caratteri.

3.6 Tabella dei volumi

ELEMENTO	TIPOLOGIA	VOLUME
Cliente	E	10000
Prenotazione	E	20000
Ricevuta	E	20000
Trattamento	E	50
Dipendente	E	10
Prodotto	E	500
Effettuare	R	20000
Utilizzare	R	20000
Eseguire	R	20000
Riguardare	R	20000

3.7 Tabella delle frequenze

* si tratta di stime approssimative

Operazioni	Descrizione	Frequenza
Op. 1	Inserimento di un nuovo cliente	50/w
Op. 2	Aggiornare il recapito telefonico di un cliente	10/y
Op. 3	Effettuare una prenotazione in base alla disponibilità	100/w
Op. 4	Contare il numero di prenotazioni effettuate in una data	12/y
Op. 5	Contare quante prenotazioni ha effettuato un cliente	100/y
Op. 6	Calcolare le entrate economiche	1/m

	mensili (guadagni dei trattamenti)	
Op. 7	Calcolare le uscite mensili dovute ai rifornimenti e agli stipendi	1/m
OP. 8	Calcolare il guadagno mensile	1/m
OP. 9	Contare quanti prodotti vengono utilizzati in un determinato trattamento	1/y

3.8 Traduzione nel modello relazionale

In grassetto: chiavi primarie

Sottolineato: chiavi esterne

Cliente(**ID_Cliente**, Nome, Cognome, Telefono)

Prenotazione(**ID_Prenotazione**, Data, Ora, ID_Cliente, ID_Trattamento)

Trattamento(**ID_Trattamento**, **ID_Prodotto**, Nome, Prezzo, Descrizione, Categoria, CF_Dipendente)

Dipendente(**CF_Dipendente**, Nome, Cognome, Telefono, Stipendio)

Prodotto(**ID_Prodotto**, Nome, Tipo, Costo)

Effettuare(**ID_Cliente**, **ID_Prenotazione**)

Riguardare(**ID_Prenotazione**, **ID_Trattamento**)

Eseguire(**ID_Dipendente**, **ID_Trattamento**)

Utilizzare(**ID_Trattamento**, **ID_Prodotto**)

4. Implementazione SQL

4.1 Tabelle

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (  
  `ID_Cliente` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Cognome` VARCHAR(45) NOT NULL,  
  `Telefono` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`ID_Cliente`),  
  UNIQUE INDEX `CF_Cliente_UNIQUE` (`ID_Cliente` ASC))  
ENGINE = MyISAM;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Dipendente` (  
  `CF_Dipendente` VARCHAR(16) NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `Cognome` VARCHAR(45) NOT NULL,  
  `Telefono` VARCHAR(45) NOT NULL,  
  `Stipendio` INT NOT NULL,  
  PRIMARY KEY (`CF_Dipendente`),  
  UNIQUE INDEX `CF_Dipendente_UNIQUE` (`CF_Dipendente` ASC))  
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Prodotto` (
```

```

`ID_Prodotto` INT NOT NULL,
`Tipo` VARCHAR(45) NOT NULL,
`Costo` INT NOT NULL,
`Nome` VARCHAR(45) NOT NULL,
PRIMARY KEY (`ID_Prodotto`),
UNIQUE INDEX `ID_Prodotto_UNIQUE` (`ID_Prodotto` ASC))
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Trattamento` (
  `ID_Trattamento` INT NOT NULL,
  `Prodotto_ID_Prodotto` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Prezzo` INT NOT NULL,
  `Descrizione` VARCHAR(45) NULL DEFAULT NULL,
  `Categoria` VARCHAR(45) NOT NULL,
  `Dipendete_CF_Dipendente` VARCHAR(16) NOT NULL DEFAULT "",
  PRIMARY KEY (`ID_Trattamento`, `Prodotto_ID_Prodotto`),
  UNIQUE INDEX `ID_Trattamento_UNIQUE` (`ID_Trattamento` ASC),
  INDEX `fk_Trattamento_Dipendete1_idx` (`Dipendete_CF_Dipendente` ASC),
  INDEX `fk_Trattamento_Prodotto1_idx` (`Prodotto_ID_Prodotto` ASC),
  UNIQUE INDEX `Categoria_UNIQUE` (`Categoria` ASC),
  CONSTRAINT `fk_Trattamento_Dipendente1`
    FOREIGN KEY (`Dipendente_CF_Dipendente`)
    REFERENCES `mydb`.`Dipendente` (`CF_Dipendente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Trattamento_Prodotto1`
    FOREIGN KEY (`Prodotto_ID_Prodotto`)

```

**REFERENCES `mydb`.`Prodotto` (`ID_Prodotto`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)**

ENGINE = InnoDB;

**CREATE TABLE IF NOT EXISTS `mydb`.`Prenotazione` (
`ID_Prenotazione` INT NOT NULL,
`Data` DATE NOT NULL,
`Ora` INT NOT NULL,
`Cliente_ID_Cliente` INT NOT NULL,
`Trattamento_ID_Trattamento` INT NOT NULL,
PRIMARY KEY (`ID_Prenotazione`),
UNIQUE INDEX `ID_Prenotazione_UNIQUE` (`ID_Prenotazione` ASC),
UNIQUE INDEX `Data_UNIQUE` (`Data` ASC),
INDEX `fk_Prenotazione_Cliente_idx` (`Cliente_ID_Cliente` ASC),
INDEX `fk_Prenotazione_Trattamento1_idx` (`Trattamento_ID_Trattamento`
ASC),
CONSTRAINT `fk_Prenotazione_Cliente`
FOREIGN KEY (`Cliente_ID_Cliente`)
REFERENCES `mydb`.`Cliente` (`ID_Cliente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Prenotazione_Trattamento1`
FOREIGN KEY (`Trattamento_ID_Trattamento`)
REFERENCES `mydb`.`Trattamento` (`ID_Trattamento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;**

4.2 Operazioni

OP. 1 *Inserimento di un nuovo cliente*

```
1.  INSERT INTO 'Cliente' ('ID_Cliente', 'Nome', 'Cognome', 'Telefono')  
VALUES ($ID_Cliente, $Nome, $Cognome, $Telefono);
```

OP. 2 *Aggiornare il recapito telefonico di un cliente*

```
2.  UPDATE Cliente  
    SET Telefono = $Telefono  
    WHERE ID_Cliente = $ID_Cliente
```

OP. 3 *Effettuare una prenotazione in base alla disponibilità*

```
3.  CREATE TRIGGER Disponibilità  
    AFTER INSERT ON Prenotazione  
    FOR EACH ROW  
    BEGIN  
        IF 3 < (Select count(*)  
                From Prenotazione P  
                Where P.Data = new.Data && P.Ora = new.Ora  
                )
```

**THEN DELETE FROM Prenotazione WHERE Id_Prenotazione =
New.Id_Prenotazione
END IF;
END;**

OP. 4 *Contare il numero di prenotazioni effettuata in una data*

**4. SELECT COUNT(*)
 FROM Prenotazione P
 WHERE P.Data = \$Data**

OP. 5 *Contare quante prenotazioni ha effettuato un cliente*

**5. Select count (*)
 From Prenotazione P, Prenotazione P2
 Where P.ID_Cliente = '\$ID_Cliente' AND P2.ID_Cliente = '\$ID_Cliente'
AND P.ID_Prenotazione <> P2.ID_Prenotazione**

OP.6 *Calcolare le entrate economiche mensili (guadagni dei trattamenti)*

**6.

Create View TrattamentoNoDoppioni(ID_Trattamento, Prezzo) AS
Select Distinct T.ID_Trattamento, T.Prezzo
From Trattamento T

Select Sum(Prezzo)
From Prenotazione P, TrattamentoNoDoppioni T**

**Where P.Data >= '\$Data' AND P.Data <= '\$Data' AND P.ID_Trattamento
= T.ID_Trattamento**

OP.7 *Calcolare le uscite mensili dovute ai rifornimenti e agli stipendi*

7. Create View Uscite AS
Select P.Costo AS 'x'
From Prodotto P
Union
Select D.Stipendio AS 'x'
From Dipendente D

Select sum(x)
from Uscite;

OP.8 *Calcolare il guadagno mensile*

8. Create View Uscite AS
Select P.Costo AS 'x'
From Prodotto P
Union
Select D.Stipendio AS 'x'
From Dipendente D

Create View UsciteMensili (Costo) AS
Select sum(x)
from Uscite

Create View TrattamentoNoDoppioni(ID_Trattamento, Prezzo) AS
Select Distinct T.ID_Trattamento, T.Prezzo
From Trattamento T

Create View EntrateMese (Ricavi) AS
Select Sum(Prezzo)
From Prenotazione P, TrattamentoNoDoppioni T
Where P.Data >= '\$Data' AND P.Data <= '\$Data' AND P.ID_Trattamento
= T.ID_Trattamento

Select (E.Ricavi - U.Costo)
From UsciteMensili U, EntrateMese E

OP.9 *Contare quanti prodotti vengono utilizzati in un determinato trattamento*

9. Select count(*)
From Trattamento T, Trattamento T2
Where T.ID_Trattamento = '\$ID_Trattamento' AND T2.ID_Trattamento =
'\$ID_Trattamento' AND T.ID_Prodotto <> T2.ID_Prodotto

4.3 Diagramma EER

