



PROGETTO DI UNA BASE DI DATI

*Database per l'organizzazione della Brew Firm TheBroda,
progetto realizzato da Gianluca Coletta e Folco Giorgetti*

PROGETTAZIONE CONCETTUALE

Analisi dei requisiti

Si vuole progettare una base di dati finalizzata a modellare l'organizzazione della Brew Firm Thebroda. Una Brew Firm è un'azienda (o un piccolo produttore) che affitta un impianto altrui avendo elaborato e costruito negli anni una ricetta propria per la birra.

Di ogni birrificio si vuole conoscere l'indirizzo, il nome e un recapito telefonico, ogni birrificio ha un proprietario di cui si vuole conoscere il codice fiscale, il nome, il cognome e un recapito telefonico.

Di ogni partita verso il birrificio si vuole conoscere il codice, la quantità di birra in litri, la data e il costo.

I clienti possono essere di due tipologie: privati e pubblici esercizi (pub, ristoranti ecc..). Del cliente privato si vuole conoscere il codice fiscale, il nome, il cognome, la data di nascita, l'indirizzo e un recapito telefonico.

Dell'esercizio pubblico si vuole conoscere il nome, il tipo, l'indirizzo, la città e un recapito telefonico; ogni esercizio pubblico ha un titolare che può essere proprietario di più esercizi della quale si vuole conoscere il codice fiscale, il nome, il cognome e un recapito telefonico.

Delle birre prodotte si vuole conoscere il nome, il tipo, il prezzo di vendita per litro, la gradazione alcolica e le materie prime utilizzate con la rispettiva quantità espressa come grammi per litro di birra.

Delle materie prime si vuole conoscere il codice, il nome, il costo al Kg e il tipo.

In ogni consegna possono essere distribuite birre di diverse tipologie; si vuole conoscere il codice, una data, un'ora, la quantità di birre e il costo totale della consegna come prodotto tra quantità di birre e rispettivo prezzo. Ad ogni consegna può partecipare un solo impiegato.

Del dipendente si vuole conoscere il nome, il cognome, il codice Fiscale, lo stipendio, un recapito telefonico.

Analisi delle specifiche

Fraasi di carattere generale:

Si vuole progettare una base di dati finalizzata a modellare l'organizzazione della Brew Firm TheBroda.

Fraasi relative ai clienti:

I clienti possono essere di due tipologie: privati e pubblici esercizi (bar, ristoranti ecc..).

- **Fraasi relative ai clienti privati:** rappresentiamo il codice fiscale, nome, cognome, data di nascita, indirizzo e un recapito telefonico.

- **Frase relative agli esercizi pubblici:** rappresentiamo il nome, tipo, indirizzo, città e un recapito telefonico; ogni esercizio pubblico ha un titolare che può essere proprietario di più esercizi.

Frase relative ai birrifici:

Di ogni birrificio rappresentiamo il nome, indirizzo e un recapito telefonico, ogni birrificio ha un proprietario di cui si vuole conoscere nome, cognome e un recapito telefonico.

Frase relative alle materie prime:

Delle materie prime rappresentiamo il codice, nome, costo e tipo.

Frase relative ai dipendenti:

Dei dipendenti rappresentiamo il nome, cognome, codice Fiscale, stipendio e un recapito telefonico.

Frase relative alle consegne:

In ogni consegna possono essere distribuite birre di diverse tipologie; rappresentiamo il codice, data, ora, quantità di birre e costo totale della consegna come prodotto tra quantità di birre e rispettivo prezzo. Ad ogni consegna possono partecipare più dipendenti.

Frase relative alle birre:

Delle birre prodotte rappresentiamo il nome, tipo, prezzo di vendita per litro, gradazione alcolica e le materie prime utilizzate con la rispettiva quantità espressa come grammi per litro di birra.

Frase relative alle partite:

Di ogni partita verso il birrificio rappresentiamo il codice, quantità di birra, la data e il Costo.

Frase relative ai titolari:

Un titolare può essere proprietario di più esercizi della quale rappresentiamo il codice fiscale, nome, cognome e un recapito telefonico.

Frase relative ai proprietari:

Di ogni proprietario rappresentiamo il codice fiscale, nome, cognome e un recapito telefonico.

Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Privato	Cliente privato che acquista delle birre.		Consegna.
Pubblico	Esercizio pubblico che acquista delle birre.		Consegna, Titolare
Birrificio	Luogo dove viene prodotta la birra.		Proprietario, Ordine
Materia	Materie prime con la quale viene prodotta la birra.		Birra.
Dipendente	Persona che effettua le consegne.	Impiegato	Consegna.
Consegna	Distribuzione fisica dei prodotti verso i vari.		Birra, Cliente, Dipendenti.
Birra	Prodotto realizzato dal birrificio.		Consegna, ordine, materie prime.
Partita	Quantità di birra richiesta al birrificio.		Birra, Birrificio.
Titolare	Persona che possiede l'esercizio pubblico.		Esercizio pubblico.
Proprietario	Persona che possiede il birrificio		Birrificio.

Elenco delle operazioni

OP1: Aggiunta di un dipendente (2 volte all'anno)

OP2: Aggiunta di un privato (1 volta al mese)

OP3: Aggiunta nuova partita (10 volte al mese)

OP4: Modifica titolare (1 volta ogni 2 anni)

OP5: Visualizzazione quantità di birra di una determinata partita (20 volte al mese)

OP6: Aggiunta nuova consegna (90 al mese)

OP7: Stampa di tutte le consegne del mese (1 volta al mese)

OP8: Eliminazione di una consegna (8 volte al mese)

OP9: Calcolo guadagno di un determinato mese (1 volta al mese)

- OP10:** Calcolo spese di un determinato anno (1 volta all'anno)
- OP11:** Visualizzazione composizione di una birra (5 volte al mese)
- OP12:** Visualizzazione del costo di una partita (10 volte mese)
- OP13:** Modifica dipendente che si occupa di una consegna (8 volte al mese)
- OP14:** Visualizzazione del proprietario di un birrificio (3 volte al mese)
- OP15:** Visualizzazione del numero di consegne effettuate dai dipendenti durante un anno (1 volta all'anno)

Schema entità-relazione

Il punto di partenza per la realizzazione del modello concettuale è il concetto di Birra che è un concetto centrale in una Brewfirm. A partire da questo si procede a macchia d'olio aggiungendo allo schema tutti gli altri concetti richiesti.



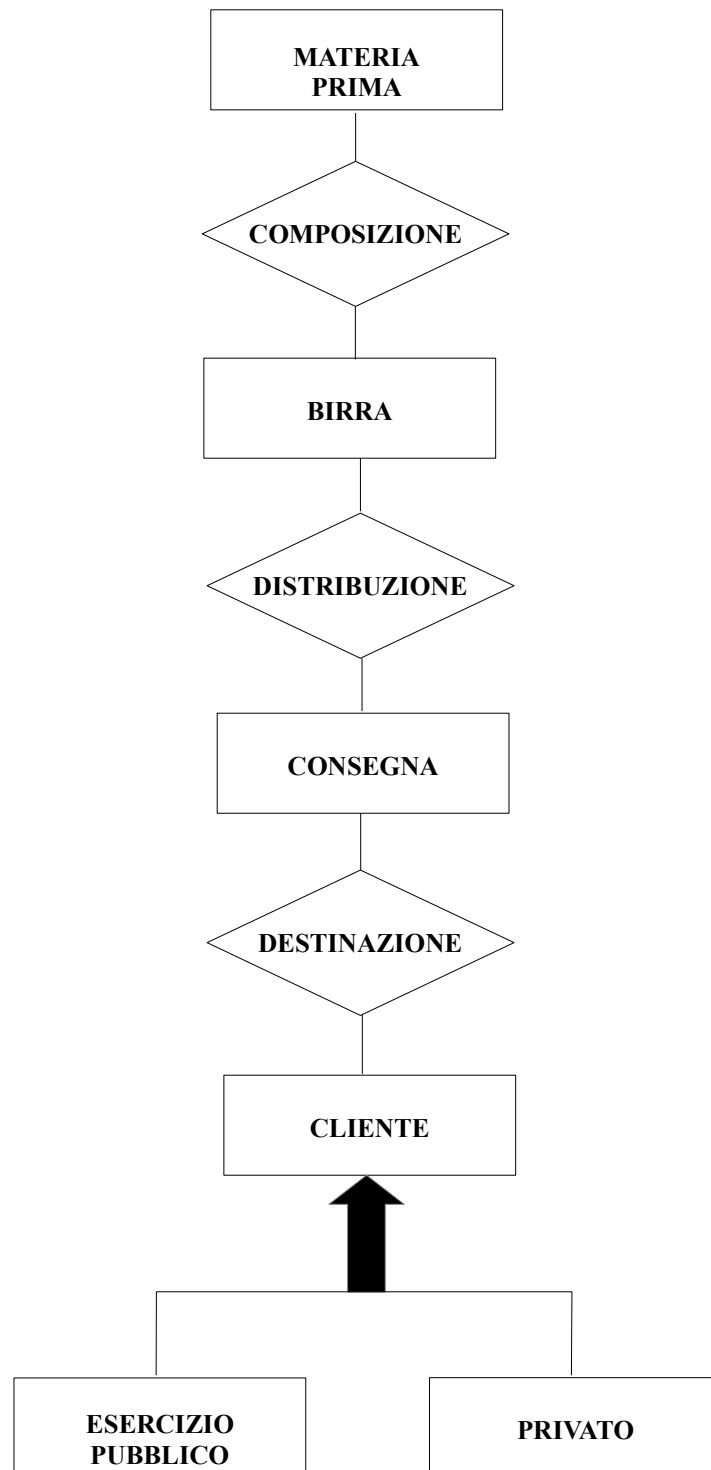
Inizialmente lo schema scheletro è composto dalla sola entità birra:

Successivamente ci interessiamo della composizione della birra cioè le materie prime utilizzate per la produzione della birra; quindi creiamo l'entità Materia Prima e la colleghiamo a Birra tramite la relazione Composizione.

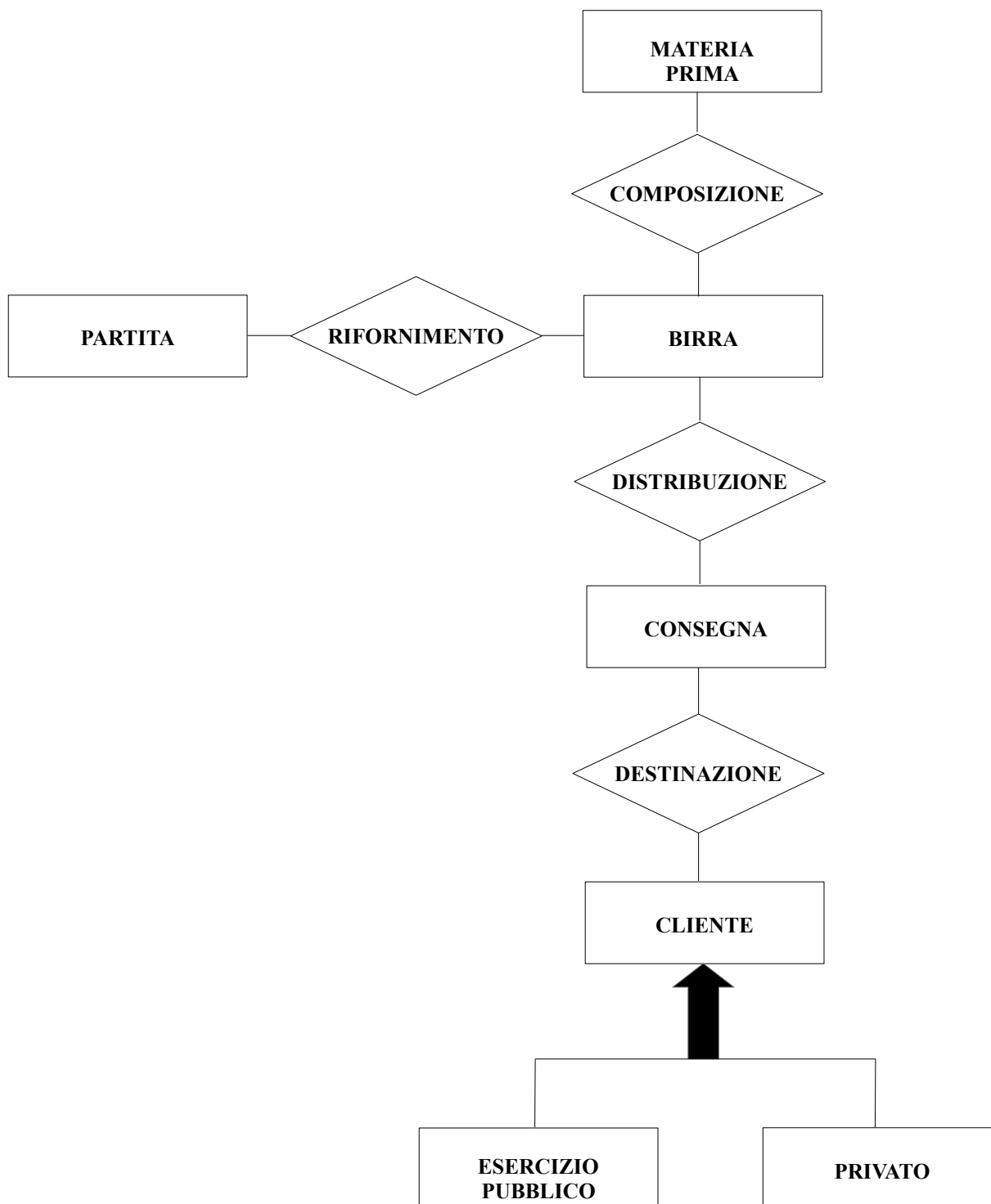


Dopodiché ci interessiamo delle consegne effettuate dall'azienda verso i clienti, aggiungendo l'entità Consegna tramite la relazione distribuzione.

Inoltre consideriamo i clienti coinvolti nelle consegne che suddividiamo tramite una generalizzazione completa in privati ed esercizi pubblici. L'entità cliente è collegata per mezzo della relazione Destinazione all'entità Consegna.



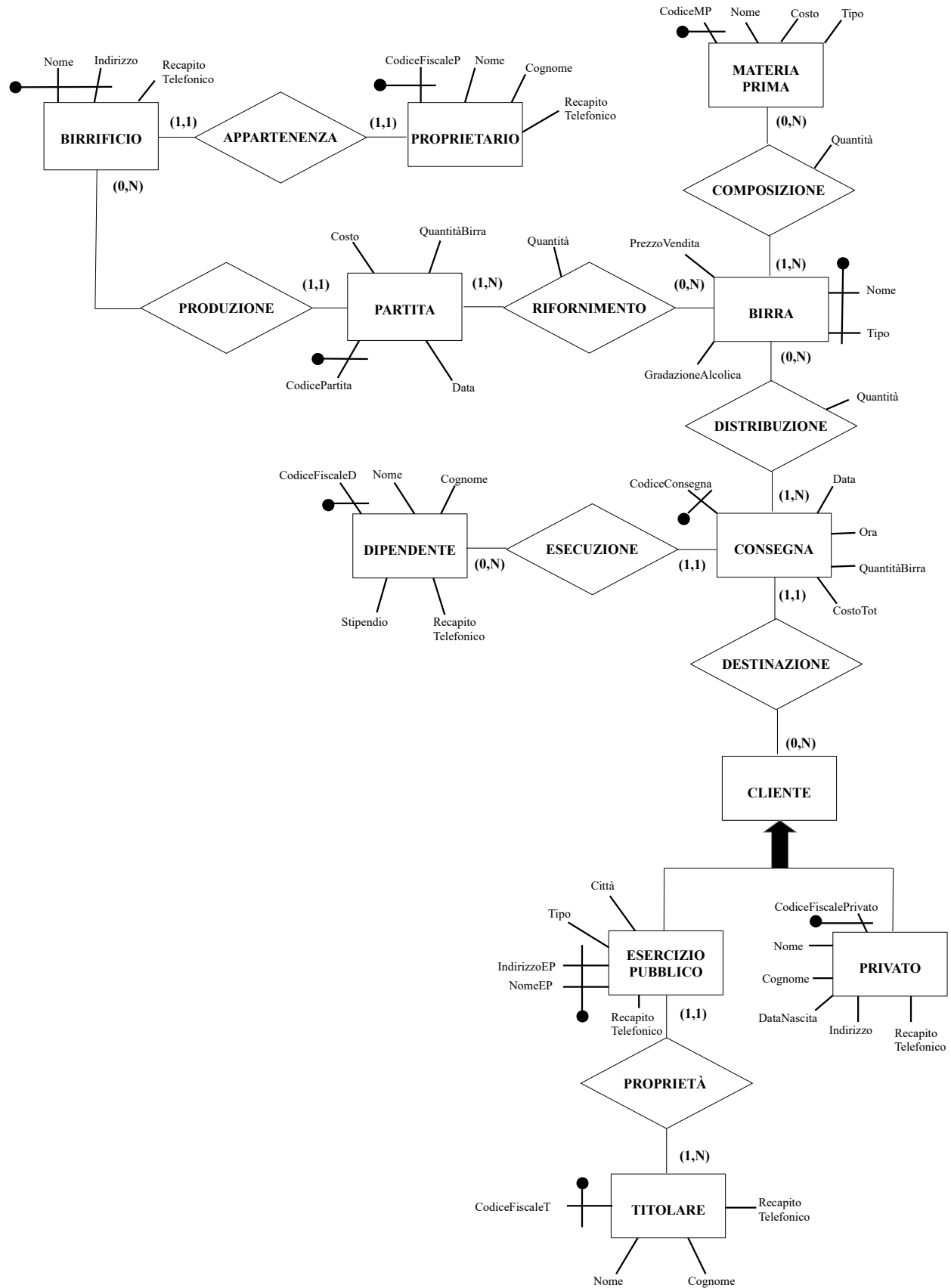
Come ultimo concetto collegato direttamente alle birre c'è quello del rifornimento di una partita di birre perciò creiamo l'entità Partita e la colleghiamo con una relazione a Birra.



Dopodiché per ogni esercizio pubblico è di interesse risalire al proprietario e quindi aggiungiamo un'entità Titolare e la leghiamo a Esercizio Pubblico tramite una relazione.

Inoltre ci interessiamo dei dipendenti coinvolti nelle consegne creando un'entità Dipendente collegata direttamente tramite una relazione a Consegna. Infine ci interessiamo del birrificio che si occuperà della partita di birre aggiungendo l'entità Birrificio collegata a Partita tramite una relazione. In più consideriamo il proprietario del birrificio creando l'entità Proprietario collegata a Birrificio tramite una relazione.

Per completare la fase di progettazione concettuale è necessario inserire gli attributi e le cardinalità delle relazioni, i quali vengono dedotti immediatamente dall'analisi delle specifiche.



Dizionario dei dati (Entità)

Entità	Identificatore	Attributi	Descrizione
PROPRIETARIO	CodiceFiscaleP	Nome, Cognome, RecapitoTelefonico	Proprietario del birrificio
BIRRIFICIO	Nome, Indirizzo	RecapitoTelefonico	Edificio dove viene prodotta la birra
PARTITA	CodicePartita	QuantitàBirra, Data, Costo	Quantità di birra ordinata
BIRRA	Nome, Tipo	GradazioneAlcolica, PrezzoVendita	Bevanda prodotta nel birrificio
MATERIA PRIMA	CodiceMP	Nome, Costo, Tipo	Materiali utilizzati per produrre la birra
DIPENDENTE	CodiceFiscaleD	Nome, Cognome, Stipendio, RecapitoTelefonico	Soggetti che lavorano per la BrewFirm
CONSEGNA	CodiceConsegna	Data, Ora, QuantitàBirra, CostoTot	Quantità di birra che la BrewFirm porta al cliente
PRIVATO	CodiceFiscalePrivato	Nome, Cognome, DataNascita, Indirizzo, RecapitoTelefonico	Soggetti che acquistano birra per uso privato
ESERCIZIO PUBBLICO	NomeEP, IndirizzoEP	Tipo, Città, RecapitoTelefonico	Locali pubblici che acquistano birra
TITOLARE	CodiceFiscaleT	Nome, Cognome, RecapitoTelefonico	Soggetto a capo dell'esercizio pubblico

Dizionario dei dati (Relazioni)

Relazione	Entità	Attributi	Descrizione
RIFORNIMENTO	Partita, Birra	Quantità	Tipo di birra richiesta
COMPOSIZIONE	Birra, MateriePrime	Quantità	Associa le materie prime alla rispettiva birra
DISTRIBUZIONE	Birra, Consegna	Quantità	Composizione della consegna
ESECUZIONE	Consegna, Dipendenti		Dipendenti addetti alla consegna
DESTINAZIONE	Consegna, Cliente		Associa una consegna al rispettivo cliente
PROPRIETA'	EserciziPubblici, Titolare		Associa un Titolare al rispettivo locale
PRODUZIONE	Birrificio, Partita		Associa il birrificio alla partita di birra prodotta
APPARTENENZA	Birrificio, Proprietario		Associa un Proprietario al rispettivo birrificio

PROGETTAZIONE LOGICA

Tavola dei volumi

Concetto	Tipo	Volume
PROPRIETARIO	E	10
BIRRIFICIO	E	10
PARTITA	E	120
BIRRA	E	18
MATERIA PRIMA	E	25
DIPENDENTE	E	8
CONSEGNA	E	1080
CLIENTE	E	34
PRIVATO	E	20
ESERCIZIO PUBBLICO	E	14
TITOLARE	E	10
RIFORNIMENTO	R	120x18=2160
COMPOSIZIONE	R	18x25=450
DISTRIBUZIONE	R	1080x18=19440
ESECUZIONE	R	1080
DESTINAZIONE	R	1080x34=36720
PROPRIETA'	R	14
PRODUZIONE	R	120
APPARTENENZA	R	10

Tavola delle operazioni

Operazione	Tipo	Frequenza
OP1: Aggiunta di un dipendente		2 volte all'anno
OP2: Aggiunta di un privato		1 volta al mese
OP3: Aggiunta nuova partita		10 volte al mese
OP4: Modifica titolare		1 volta ogni 2 anni
OP5: Visualizzazione quantità di birra di una determinata partita		20 volte al mese
OP6: Aggiunta nuova consegna		90 al mese
OP7: Stampa di tutte le consegne del mese		1 volta al mese
OP8: Eliminazione di una consegna		8 volte al mese
OP9: Calcolo guadagno di un determinato mese		1 volta al mese
OP10: Calcolo spese di un determinato anno		1 volta all'anno
OP11: Visualizzazione composizione di una birra		5 volte al mese
OP12: Visualizzazione del costo di una partita.		10 volte mese
OP13: Modifica dipendente che si occupa di una consegna		8 volte al mese
OP14: Visualizzazione del proprietario di un birrificio		3 volte al mese
OP15: Visualizzazione del numero di consegne effettuate dai dipendenti durante un anno		1 volta all'anno

Analisi delle ridondanze

Una possibile ridondanza è quella relativa all'attributo CostoTot dell'entità Consegna. Il costo totale di una consegna si può infatti calcolare come il prodotto tra la quantità di birre della consegna per il rispettivo prezzo.

Per determinare se conviene o meno mantenere questa ridondanza occorre analizzare e quindi confrontare gli indici di prestazione nei due casi, con e senza ridondanza. L'indice di prestazione viene calcolato sulla base delle operazioni che coinvolgono relazioni ed entità, oggetto della ridondanza.

L'operazione per la quale devo calcolare gli indici di prestazione in questo caso è:

OP9: Calcolo guadagno di un determinato mese (1 volta al mese)



Se mantengo la ridondanza e quindi l'attributo CostoTot di Consegna devo effettuare un solo accesso in lettura 90 volte al mese.

Costo con ridondanza: $1L \times 90 = 90$ letture al mese.

Poiché il costo massimo di una consegna non supera quasi mai i 1000€ posso usare 2Byte per rappresentarlo calcolo quindi la memoria occupata dall'attributo CostoTot come prodotto tra la dimensione dell'attributo e la dimensione dell'entità:

$2 \text{ Byte} \times 1080 \text{ consegne} = 2160 \text{ Byte} = 2.16 \text{ Kbyte}$

Se invece elimino l'attributo ridondante devo effettuare una lettura su Birra, una su Consegna e una sulla relazione Distribuzione per ogni tipo di birra della consegna, che in media è composta da 3 tipi diversi di birra. Tutto ciò va fatto 90 volte al mese.

Costo senza ridondanza: $3L \times 3(\text{tipi}) \times 90 = 810$ letture al mese.

Poiché il costo senza ridondanza è 9 volte maggiore di quello con ridondanza conviene mantenere la ridondanza e usare 2.16Kbyte di memoria per mantenere l'attributo CostoTot.

Un'altra ridondanza è data dall'attributo QuantitàBirra dell'entità Partita che può essere calcolata come la somma delle rispettive quantità di ogni tipo di birra tramite l'attributo quantità della relazione Rifornimento.

Per determinare se conviene o meno mantenere questa ridondanza occorre analizzare e quindi confrontare gli indici di prestazione nei due casi, con e senza ridondanza. L'indice di prestazione viene calcolato sulla base delle operazioni che coinvolgono relazioni ed entità, oggetto della ridondanza.

L'operazione per la quale devo calcolare gli indici di prestazione in questo caso è:

OP5: Visualizzazione quantità di birra di una determinata partita (20 volte al mese).



Se mantengo la ridondanza e quindi l'attributo QuantitàBirra di Partita devo effettuare un solo accesso in lettura 20 volte al mese.

Costo con ridondanza: $1L \times 20 = 20$ letture al mese.

Poiché la quantità di birra massima di una partita non supera quasi mai i 1000L posso usare 2Byte per rappresentarlo calcolo quindi la memoria occupata dall'attributo QuantitàBirra come prodotto tra la dimensione dell'attributo e la dimensione dell'entità:

$2 \text{ Byte} \times 120 \text{ partite} = 240 \text{ Byte}$

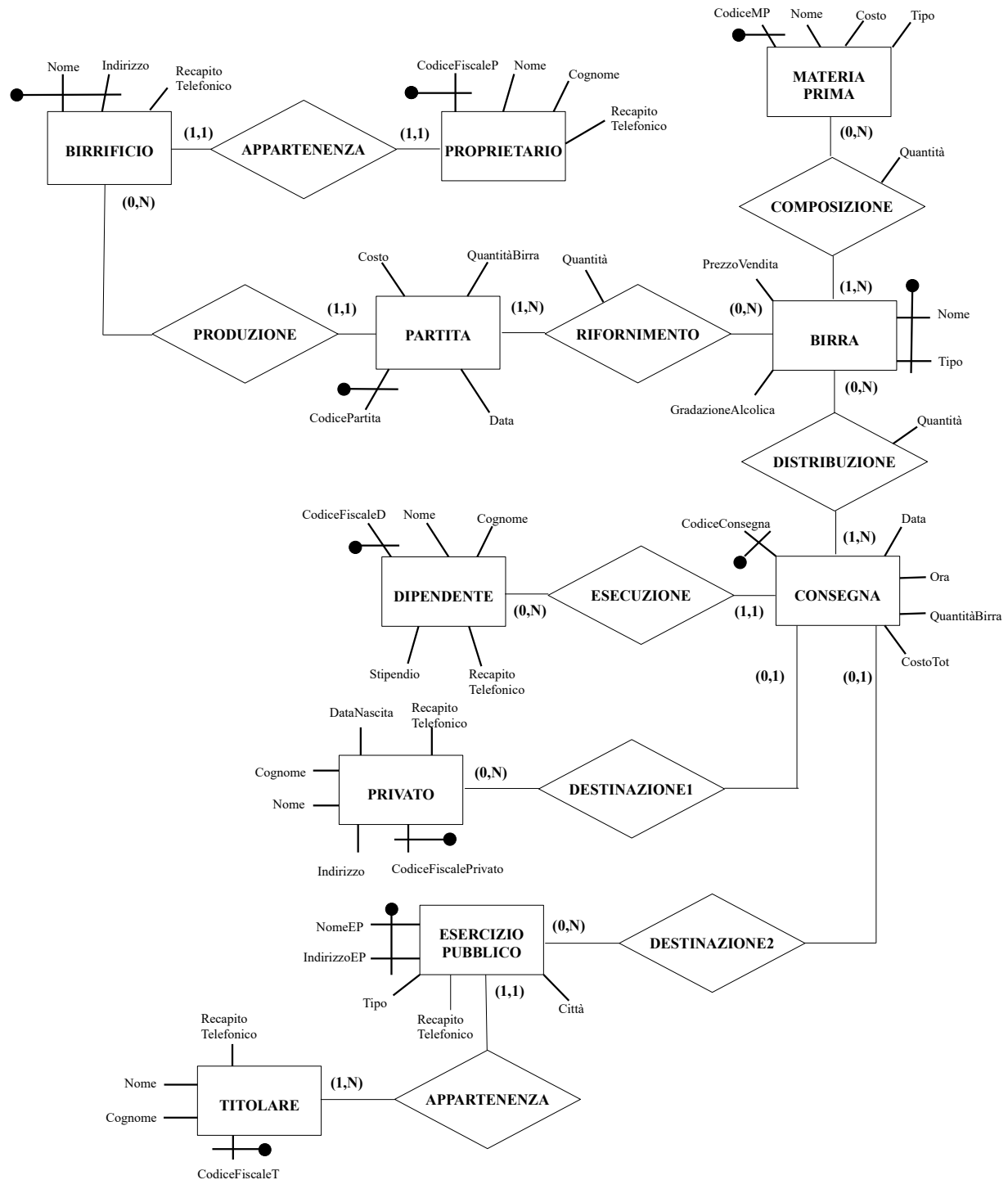
Se invece elimino l'attributo ridondante devo effettuare solamente una lettura sulla relazione Rifornimento 20 volte al mese per ogni partita, composta in media da 2 tipi di birre diverse.

Costo senza ridondanza: $1L \times 2(\text{tipi}) \times 20 = 40$ letture al mese.

Poiché l'attributo QuantitàBirra occupa poco spazio in memoria ed essendo il costo senza ridondanza è il doppio di quello con ridondanza conviene usare 240 Byte di memoria per mantenere l'attributo.

Eliminazione di generalizzazioni

Nello schema è presente una generalizzazione sull'entità CLIENTI. La generalizzazione è di tipo totale ed esclusivo, inoltre per l'entità figlia ESERCIZIO PUBBLICO ci interessano le informazioni sul Titolare dell'esercizio. Decido quindi di mantenere soltanto le entità figlie trasferendo in esse gli attributi dell'entità CLIENTE.



Scelta degli identificatori primari

La scelta degli identificatori primari è stata fatta con i seguenti criteri:

- Per le entità Proprietario, Titolare, Privato e Dipendente si sono scelti come identificatori primari rispettivamente gli attributi CodiceFiscaleP, CodiceFiscaleT, CodiceFiscalePrivato, CodiceFiscaleD, che sono già identificativi di ogni persona.
- Per le entità Birrificio ed Esercizio Pubblico si scelgono Nome e Indirizzo assumendo che non esistano due locali con lo stesso nome e lo stesso indirizzo.
- Per le entità Materia Prima, Consegna e Partita si scelgono rispettivamente CodiceMP, CodiceConsegna e CodicePartita, numeri interi assegnati dall'azienda con lo scopo di rendere univoca ogni consegna, ogni materia prima e ogni partita.
- Per l'entità Birra si scelgono Nome e Tipo assumendo che non esistano due birre dello stesso tipo e con lo stesso nome.

Modello Relazionale

Traduzione delle entità:

PROPRIETARIO(CodiceFiscaleP, Nome, Cognome, RecapitoTelefonico);

BIRRIFICIO (Nome, Indirizzo, RecapitoTelefonico, Proprietario);

Con vincolo di integrità referenziale tra l'attributo Proprietario e la relazione Proprietario.

PARTITA(CodicePartita, QuantitàBirra, Data, Costo, NomeBirrificio, IndirizzoBirrificio);

Con vincolo di integrità referenziale tra gli attributi NomeBirrificio, IndirizzoBirrificio e la relazione Birrificio.

BIRRA(Nome, Tipo, GradazioneAlcolica, PrezzoVendita);

MATERIA PRIMA(CodiceMP, Nome, Costo, Tipo);

DIPENDENTE(CodiceFiscaleD, Nome, Cognome, Stipendio, RecapitoTelefonico);

CONSEGNA(CodiceConsegna, Data, Ora, CostoTot, QuantitàBirra, Dipendente);

Con vincolo di integrità referenziale tra l'attributo Dipendente e la relazione Dipendente.

PRIVATO(CodiceFiscalePrivato, Nome, Cognome, DataNascita, Indirizzo, RecapitoTelefonico);

TITOLARE(CodiceFiscaleT, Nome, Cognome, RecapitoTelefonico)

ESERCIZIO PUBBLICO(NomeEP, IndirizzoEP, Tipo, Città, RecapitoTelefonico, Titolare);

Con vincolo di integrità referenziale tra l'attributo Titolare e la relazione Titolare.

Traduzione delle relazioni:

RIFORNIMENTO(NomeBirra, TipoBirra, Partita, Quantità);

Con vincolo di integrità referenziale tra l'attributo Partita e la relazione Partita e tra gli attributi NomeBirra, TipoBirra e la relazione Birra.

COMPOSIZIONE(MateriaPrima, NomeBirra, TipoBirra, Quantità);

Con vincolo di integrità referenziale tra l'attributo MateriaPrima e la relazione Materia Prima e tra gli attributi NomeBirra, TipoBirra e la relazione Birra.

DISTRIBUZIONE(Consegna, NomeBirra, TipoBirra, Quantità);

Con vincolo di integrità referenziale tra l'attributo Consegna e la relazione Consegna e tra gli attributi NomeBirra, TipoBirra e la relazione Birra.

DESTINAZIONE1(Privato, Consegna);

Con vincolo di integrità referenziale tra l'attributo Consegna e la relazione Consegna e tra l'attributo Privato e la relazione Privato.

DESTINAZIONE2(NomeEsercizio, IndirizzoEsercizio, Consegna);

Con vincolo di integrità referenziale tra l'attributo Consegna e la relazione Consegna e tra gli attributi NomeEsercizio, IndirizzoEsercizio e la relazione EsercizioPubblico.

IMPLEMENTAZIONE DELLE OPERAZIONI

Creazione delle tabelle

CREATE TABLE Proprietario

```
(  
    CodiceFiscaleP character(16) NOT NULL,  
    Nome character varying(20),  
    Cognome character varying(20),  
    RecapitoTelefonico integer,  
    CONSTRAINT "Proprietario_pkey" PRIMARY KEY (CodiceFiscaleP)  
)
```

CREATE TABLE Birrificio

```
(  
    Nome character varying(30) NOT NULL,  
    Indirizzo character varying(50) NOT NULL,  
    RecapitoTelefonico integer,  
    Proprietario character varying(16),  
    CONSTRAINT "Birrificio_pkey" PRIMARY KEY (Nome, Indirizzo),  
    CONSTRAINT "Birrificio_Proprietario_fkey" FOREIGN KEY (Proprietario)  
    REFERENCES Proprietario (CodiceFiscaleP)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Partita

```
(  
    CodicePartita integer NOT NULL,  
    QuantitàBirra real,
```

Data date,
Costo real,
NomeBirrificio character varying(30),
IndirizzoBirrificio character varying(50),
CONSTRAINT "Partita_pkey" PRIMARY KEY (CodicePartita),
CONSTRAINT "Partita_NomeBirrificioIndirizzoBirrificio_fkey" FOREIGN KEY
(NomeBirrificio, IndirizzoBirrificio)
REFERENCES Birrificio (Nome, Indirizzo)
ON UPDATE CASCADE ON DELETE CASCADE
)

CREATE TABLE Birra

(
Nome character varying(20) NOT NULL,
Tipo character varying(30) NOT NULL,
GradazioneAlcolica real,
PrezzoVendita real,
CONSTRAINT "Birra_pkey" PRIMARY KEY (Nome, Tipo)
)

CREATE TABLE MateriaPrima

(
CodiceMP integer NOT NULL,
Tipo character varying(30),
Costo real,
Nome character varying(20),
CONSTRAINT "MateriaPrima_pkey" PRIMARY KEY (CodiceMP)
)

CREATE TABLE Dipendente

```
(  
    CodiceFiscaleD character varying(16) NOT NULL,  
    Nome character varying(20),  
    Cognome character varying(20),  
    Stipendio real,  
    RecapitoTelefonico integer,  
    CONSTRAINT "Dipendente_pkey" PRIMARY KEY (CodiceFiscaleD)  
)
```

CREATE TABLE Consegna

```
(  
    CodiceConsegna integer NOT NULL,  
    Data date,  
    Ora time,  
    CostoTot real,  
    QuantitàBirra real,  
    Dipendente character varying(16),  
    CONSTRAINT "Consegna_pkey" PRIMARY KEY (CodiceConsegna),  
    CONSTRAINT "Consegna_Dipendente_fkey" FOREIGN KEY (Dipendente)  
        REFERENCES Dipendente (CodiceFiscaleD)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Privato

```
(  
    CodiceFiscalePrivato character varying(16) NOT NULL,  
    Nome character varying(20),
```

```
Cognome character varying(20),  
DataNascita date,  
Indirizzo character varying(50),  
RecapitoTelefonico integer,  
CONSTRAINT "Privato_pkey" PRIMARY KEY (CodiceFiscalePrivato)  
)
```

CREATE TABLE Titolare

```
(  
CodiceFiscaleT character varying(16) NOT NULL,  
Nome character varying(20),  
Cognome character varying(20),  
RecapitoTelefonico integer,  
CONSTRAINT "Titolare_pkey" PRIMARY KEY (CodiceFiscaleT)  
)
```

CREATE TABLE EsercizioPubblico

```
(  
NomeEP character varying(30) NOT NULL,  
IndirizzoEP character varying(50) NOT NULL,  
Tipo character varying(20),  
Città character varying(20),  
RecapitoTelefonico integer,  
Titolare character varying(16),  
CONSTRAINT "EsercizioPubblico_pkey" PRIMARY KEY (NomeEP, IndirizzoEP),  
CONSTRAINT "EsercizioPubblico_Titolare_fkey" FOREIGN KEY (Titolare)  
REFERENCES Titolare (CodiceFiscaleT)  
ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Rifornamento

```
(  
    NomeBirra character varying(20) NOT NULL,  
    TipoBirra character varying(30) NOT NULL,  
    Partita integer NOT NULL,  
    Quantità real,  
    CONSTRAINT "Rifornimento_pkey" PRIMARY KEY (NomeBirra, TipoBirra, Partita),  
    CONSTRAINT "Rifornimento_NomeBirraTipoBirra_fkey" FOREIGN KEY (NomeBirra,  
    TipoBirra)  
    REFERENCES Birra (Nome, Tipo),  
    CONSTRAINT "Rifornimento_Partita_fkey" FOREIGN KEY (Partita)  
    REFERENCES Partita (CodicePartita)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Composizione

```
(  
    NomeBirra character varying(20) NOT NULL,  
    TipoBirra character varying(30) NOT NULL,  
    MateriaPrima integer NOT NULL,  
    Quantità real,  
    CONSTRAINT "Composizione_pkey" PRIMARY KEY (NomeBirra, TipoBirra, MateriaPrima),  
    CONSTRAINT "Composizione_NomeBirraTipoBirra_fkey" FOREIGN KEY (NomeBirra,  
    TipoBirra)  
    REFERENCES Birra (Nome, Tipo),  
    CONSTRAINT "Composizione_MateriaPrima_fkey" FOREIGN KEY (MateriaPrima)  
    REFERENCES MateriaPrima (CodiceMP)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Distribuzione

```
(  
    NomeBirra character varying(20) NOT NULL,  
    TipoBirra character varying(30) NOT NULL,  
    Consegna integer NOT NULL,  
    Quantità real,  
    CONSTRAINT "Distribuzione_pkey" PRIMARY KEY (NomeBirra, TipoBirra, Consegna),  
    CONSTRAINT "Distribuzione_NomeBirraTipoBirra_fkey" FOREIGN KEY (NomeBirra,  
    TipoBirra)  
    REFERENCES Birra (Nome, Tipo),  
    CONSTRAINT "Distribuzione_Consegna_fkey" FOREIGN KEY (Consegna)  
    REFERENCES Consegna (CodiceConsegna)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```

CREATE TABLE Destinazione1

```
(  
    Privato character varying(16) NOT NULL,  
    Consegna integer NOT NULL,  
    CONSTRAINT "Destinazione1_pkey" PRIMARY KEY (Privato, Consegna),  
    CONSTRAINT "Destinazione1_Privato_fkey" FOREIGN KEY (Privato)  
    REFERENCES Privato (CodiceFiscalePrivato),  
    CONSTRAINT "Destinazione1_Consegna_fkey" FOREIGN KEY (Consegna)  
    REFERENCES Consegna (CodiceConsegna)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)
```


CREATE TABLE Destinazione2

(

NomeEsercizio character varying(30) NOT NULL,

IndirizzoEsercizio character varying(50) NOT NULL ,

Consegna integer NOT NULL,

CONSTRAINT "Destinazione2_pkey" PRIMARY KEY (NomeEsercizio,
IndirizzoEsercizio, Consegna),

CONSTRAINT "Destinazione2_NomeEsercizioIndirizzoEsercizio_fkey"
FOREIGN KEY (NomeEsercizio, IndirizzoEsercizio)

REFERENCES EsercizioPubblico (NomeEP, IndirizzoEP),

CONSTRAINT "Destinazione2_Consegna_fkey" FOREIGN KEY (Consegna)

REFERENCES Consegna (CodiceConsegna)

ON UPDATE CASCADE ON DELETE CASCADE

)

Creazione delle query

OP1: Aggiunta di un dipendente (2 volte all'anno)

```
INSERT INTO Dipendente(CodiceFiscaleD, Nome, Cognome, Stipendio, RecapitoTelefonico)
VALUES ('PSCGNN64B01A475J', 'Gianni', 'Pesci', '600', '343434031')
```

OP2: Aggiunta di un privato(1 volta al mese)

```
INSERT INTO Privato(CodiceFiscalePrivato, Nome, Cognome, DataNascita, Indirizzo,
RecapitoTelefonico)
VALUES('CRTGMR45A01G478D', 'Gianmarco', 'Curti', '1945-1-1', 'Via XX Settembre 24',
'332313103')
```

OP3: Aggiunta nuova partita (10 volte al mese)

```
INSERT INTO Partita(CodicePartita, QuantitàBirra, Data, Costo, NomeBirrifificio,
IndirizzoBirrifificio)
VALUES('9993', '650', '2019-1-2', '1350', 'La Gramigna', 'Via Eugubina 2')
```

Per inserire una nuova tupla Partita gli attributi NomeBirrifificio e IndirizzoBirrifificio devono essere già delle occorrenze nelle rispettive tabelle.

OP4: Modifica titolare (1 volta ogni 2 anni)

```
UPDATE Titolare SET RecapitoTelefonico = '345556435' WHERE CodiceFiscaleT =
'DFSHRI45D34H653L'
```

OP5: Visualizzazione quantità di birra di una determinata partita (20 volte al mese)

```
SELECT QuantitàBirra
```

```
FROM Partita
```

```
WHERE CodicePartita='4986'
```

OP6: Aggiunta nuova consegna (90 al mese)

```
INSERT INTO Consegna(CodiceConsegna, Data, Ora, CostoTot, QuantitàBirra, Dipendente)
VALUES('99843', '2019-4-5', '12:00:00', '660', '60', 'GLRDNA43L23L792L')
```

Per inserire una nuova tupla Consegna l'attributo Dipendente deve essere già un'occorrenza nella rispettiva tabella.

OP7: Stampa di tutte le consegne del mese (1 volta al mese)

```
SELECT *
```

```
FROM Consegna
```

```
WHERE Data >= '2019-02-01' and Data<='2019-02-28'
```

OP8: Eliminazione di una consegna (8 volte al mese)

```
DELETE FROM Consegna WHERE CodiceConsegna = '26523'
```

OP9: Calcolo guadagno di un determinato mese (1 volta al mese)

```
CREATE VIEW Uscite(Uscite) AS
```

```
SELECT SUM(Costo)
```

```
FROM Partita p
```

```
WHERE p.data >= '2019-02-01' AND p.data <= '2019-02-28'
```

```
SELECT SUM(CostoTot)-Uscite AS Guadagno
```

```
FROM Consegna C, Uscite
```

```
WHERE C.Data >= '2019-02-01' and C.Data<='2019-02-28'
```

```
GROUP BY Uscite
```

OP10: Calcolo spese di un determinato anno (1 volta all'anno)

```
SELECT SUM(Costo)
```

```
FROM Partita
```

```
WHERE Data>='2019-01-01' and Data<='2019-12-31'
```

OP11: Visualizzazione composizione di una birra (5 volte al mese)

```
SELECT b.Nome, m.Nome, m.Tipo, c.quantità
```

```
FROM MateriaPrima m, Birra b, Composizione c
```

```
WHERE m.codiceMP=c.MateriaPrima and b.nome=c.nomebirra and b.tipo=c.TipoBirra and  
nomebirra='Biffo'
```

OP12: Visualizzazione del costo di una partita. (10 volte mese)

```
SELECT Costo
```

```
FROM Partita
```

```
WHERE CodicePartita='1111'
```

OP13: Modifica dipendente che si occupa di una consegna (8 volte al mese)

```
UPDATE Consegna SET Dipendente='RFSLEO34O12O476L' WHERE CodiceConsegna='35264'
```

OP14: Visualizzazione del proprietario di un birrificio (3 volte al mese)

```
SELECT p.Nome, p.Cognome, p.RecapitoTelefonico
```

```
FROM Proprietario p, Birrificio b
```

```
WHERE p.CodiceFiscaleP=b.Proprietario and b.Nome= 'Birra Perugia'
```

OP15: Visualizzazione del numero di consegne effettuate dai dipendenti durante un anno (1 volta all'anno)

```
SELECT d.nome, d.cognome, codicefiscaled, COUNT(codiceconsegna)
```

```
FROM Dipendente d, Consegna c
```

```
WHERE codicefiscaled=c.dipendente and c.data>= '2019-01-01' and c.data<= '2019-12-31'
```

```
GROUP BY codicefiscaled
```