



Progetto Basi di Dati 2021/22



UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Realizzato da:

Maurizio Speranza 0124002054

Andrea Aristarco 0124002166

Progettazione	3
1.1 Sintesi dei requisiti	3
1.2 Glossario	3
1.3 Diagramma ER	4
1.4 Diagramma relazionale	4
1.5 Utenti e categorie	5
1.6 Operazioni degli utenti	5
1.7 Volumi	8
1.8 Vincoli d'integrità	8
1.9 Verifica di normalità	9
1.9.1 Prima forma normale	9
1.9.2 Seconda forma normale	9
1.9.3 Terza forma normale	9
Implementazione	10
2.1 Creazione utenti	10
2.2 Data Definition Language (DDL)	10
2.2.1 Cliente	10
2.2.2 Fumetto	11
2.2.3 Abbonamento	11
2.2.4 Magazzino	11
2.2.5 Casa editrice	11
2.2.6 Fornitore	12
2.2.7 Fornitura	12
2.2.8 Personale	12
2.2.9 Turno effettuato	12
2.2.10 Turno programmato	13
2.2.11 Evento	13
2.2.12 Stipendio	13
2.2.13 Si iscrive	13
2.2.14 Acquista	14
2.2.15 Prenota	14
2.2.16 Fornisce	14
2.2.17 Spedisce	14
2.3 Data Manipulation Language (DML)	15
2.3.1 Cliente	15
2.3.2 Fumetto	15
2.3.3 Abbonamento	16

2.3.4 Magazzino	16
2.3.5 Casa editrice	17
2.3.6 Fornitore	17
2.3.7 Fornitura	17
2.3.8 Personale	18
2.3.9 Turno effettuato	18
2.3.10 Turno programmato	18
2.3.11 Evento	18
2.3.12 Stipendio	19
2.3.13 Si iscrive	19
2.3.14 Acquista	20
2.3.15 Prenota	20
2.3.16 Fornisce	21
2.3.17 Spedisce	21
3.3 Trigger	22
3.3.1 Salario Personale	22
3.3.2 Variazione Stipendio	22
3.3.3 Controllo data turno	23
3.3.4 Capienza evento	23
3.3.5 Controllo Fornitura	24
3.3.6 Data scadenza	24
3.4 Procedure	24
3.4.1 Tipo salario	25
3.4.2 Stampa contatti	25
3.4.3 Effettua sconto	25
3.4.4 Cambio mansione	26

1. Progettazione

1.1 Sintesi dei requisiti

L'idea del database riguarda la gestione di una fumetteria la quale presente diversi servizi, tra i quali :

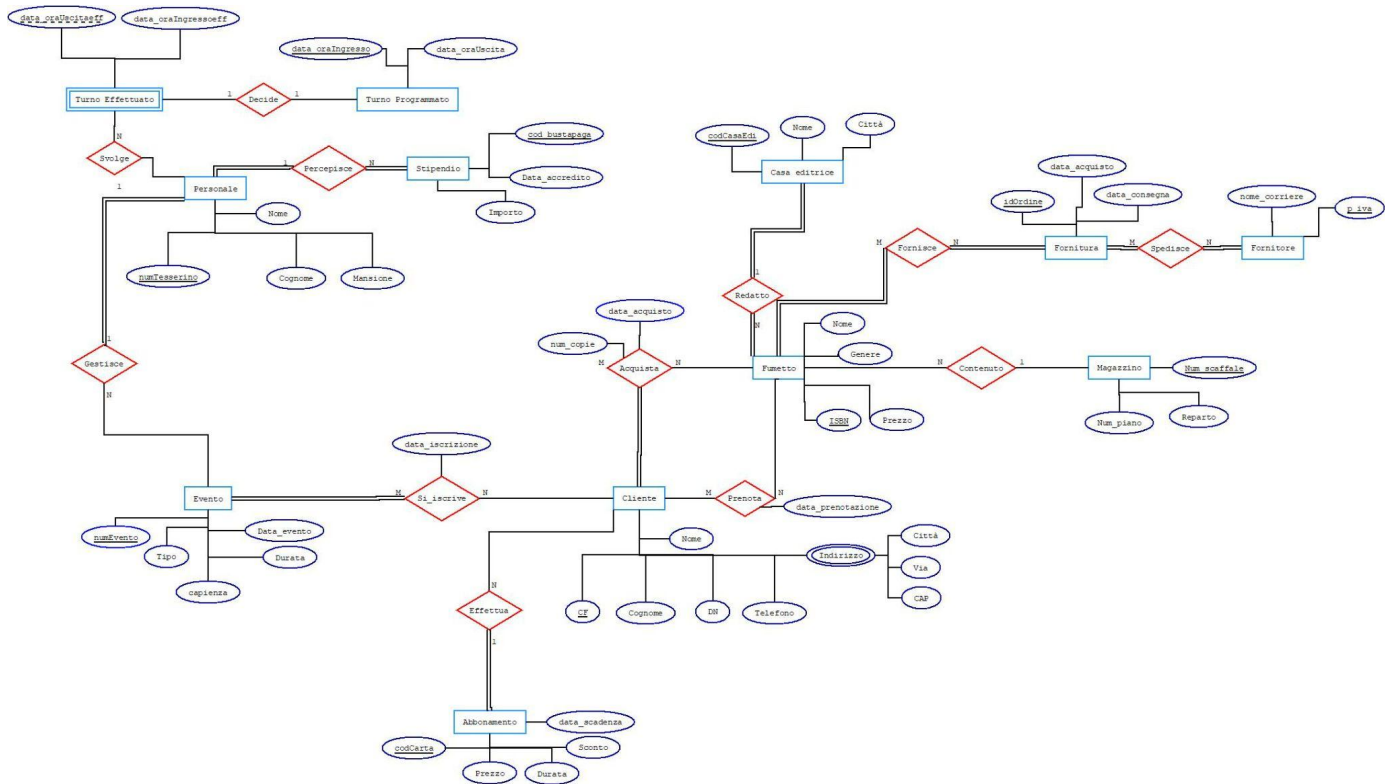
- Vendita di fumetti;
- Possibilità di prenotazione di fumetti ;
- Gestione del personale, dei propri turni lavorativi ;
- Gestione del magazzino in cui vengono depositati i fumetti;

1.2 Glossario

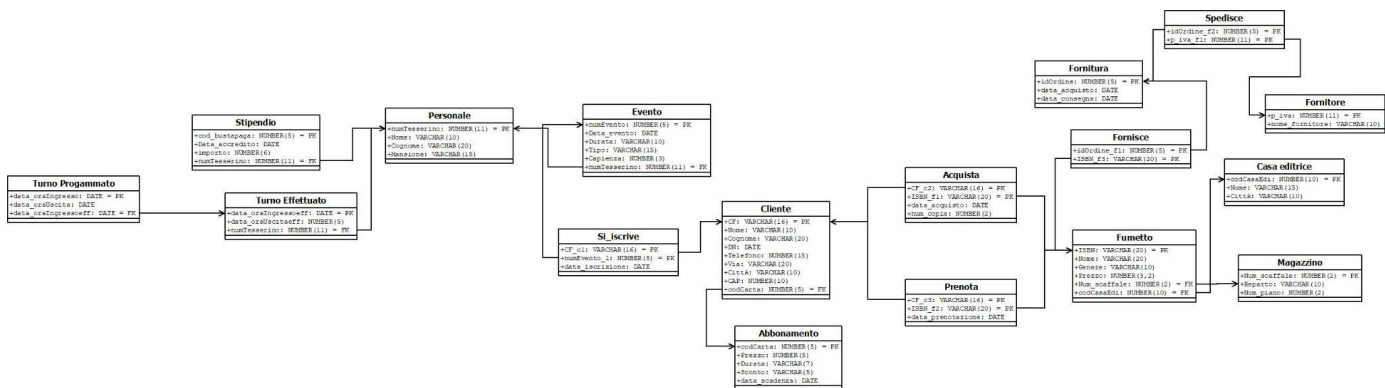
All'interno del database sono di particolare interesse le informazioni relative a:

- Fumetto identificato da un ISBN, un nome, un genere ed un prezzo;
- Cliente identificato con CF, Nome, Cognome, Data di nascita, Telefono e Indirizzo
- Magazzino identificato con il numero scaffale, reparto e un piano
- Turno effettuato identificato con data e ora d'ingresso e di uscita effettuata
- Turno programmato identificato con data e ora d'ingresso e di uscita programmata
- Personale identificato con un numero tesserino, nome. cognome e mansione
- Stipendio identificato con il codice busta paga, l'importo e la data dell'accredito
- Abbonamento identificato con un codice carta, prezzo, durata e sconto
- Fornitore identificato con una partita iva e il nome fornitore
- Fornitura identificata con un idOrdine, una data d'acquisto e data_consegna
- Casa editrice identificata con un codice, un nome e la città.
- Evento identificato con il numero, la tipologia, la capienza massima, la data e la durata

1.3 Diagramma ER



1.4 Diagramma relazionale



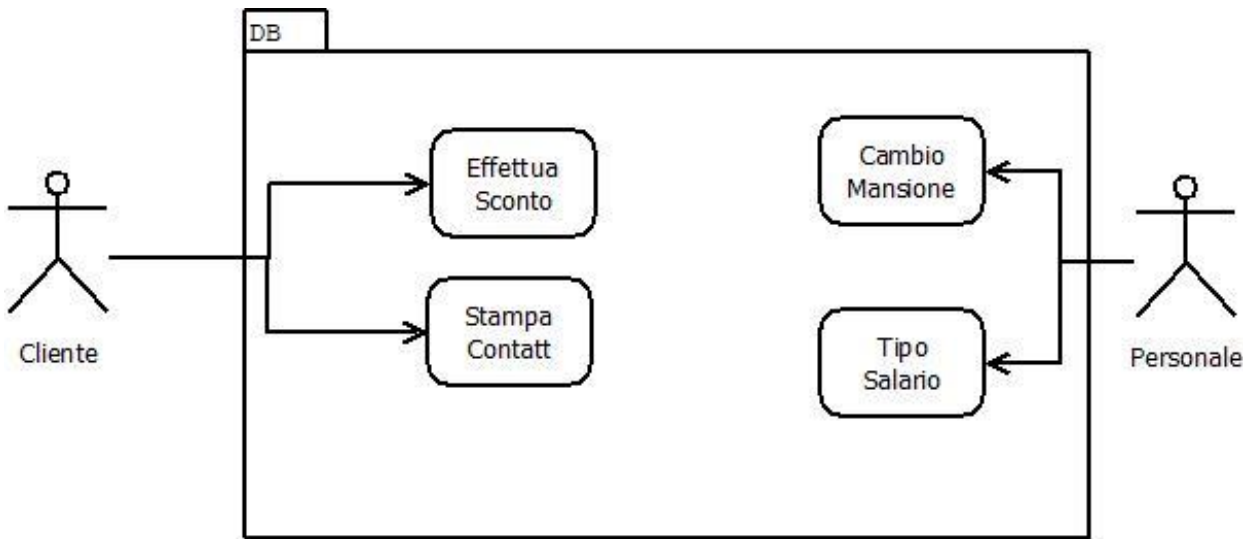


1.5 Utenti e categorie

UTENTE	TIPO	VOLUME	PERMESSI
DB_starshop	Amministratore	1	ALL
Cliente	Comune	1	EXECUTE Effettua Sconto EXECUTE Stampa Contatto
Personale	Comune	1	EXECUTE Cambio Mansione EXECUTE Tipo Salario

1.6 Operazioni degli utenti

Le operazioni di base sono quelle eseguibili con un semplice comando DML. In questo caso sono l’inserimento, l’aggiornamento o la modifica di una tupla di Prenotazione da parte del cliente. Le operazioni degli utenti sono quelle più complesse che vanno implementate tramite procedure, e in questo caso sono tipo salario, stampa contatto, effettua sconto e cambio mansione.



Seguono le schede, descrittore operazione, per le operazioni degli utenti:



Stampa Contatto:

OPERAZIONE	Stampa contatto
Scopo	Stampare info cliente dotato di abbonamento
Argomenti	numerocodCarta
Risultato	Form con dati
Errori:	Cliente sprovvisto di abbonamento
Usa	Abbonamento
Modifica	Abbonamento
Prima	Cliente si abbona
Dopo	Stampa informazioni

Effettua Sconto:

OPERAZIONE	Effettua Sconto
Scopo	Applicare lo sconto del 5% in caso di utente provvisto di abbonamento
Argomenti	numCarta, ISBN1
Risultato	Sconto del 5%
Errori:	In caso di cliente sprovvisto di abbonamento, non viene applicato lo sconto
Usa	Abbonamento
Modifica	Fumetto
Prima	Il prezzo del fumetto è base
Dopo	Il prezzo viene scontato



Cambio Mansione

OPERAZIONE	Cambio Mansione
Scopo	Cambiare mansione e importo dello stipendio ad un impiegato
Argomenti	codicetess, ruolo
Risultato	Aggiornamento del ruolo e dell'importo dell'impiegato
Errori:	In caso di impiegato non assunto, presenta l'errore
Usa	Personale
Modifica	Personale, Importo
Prima	L'impiegato ha una mansione
Dopo	L'impiegato cambia mansione e importo

Tipo Salario:

OPERAZIONE	Tipo Salario
Scopo	In base all'importo dato in input, viene mostrata la mansione assegnata
Argomenti	numtessera, importo
Risultato	Stampa delle generalità dell'impiegato e la mansione assegnatagli
Errori:	Importo inserito errato
Usa	Personale
Modifica	Personale
Prima	L'impiegato non ha alcuna mansione
Dopo	viene assegnata la mansione all'impiegato



1.7 Volumi

La tavola dei volumi è rappresentata nella tabella. Oltre a riportare il numero verosimile di tuple presenti in ciascuna tabella una volta che il DB sia a regime, rappresenta anche il suo incremento atteso in un periodo di tempo prefissato.

TABELLA	TIPO	VOLUME	INCREMENTO	PERIODO
Turno Programmato	E	14	7	Mese
Turno Effettuato	ED	14	7	Mese
Personale	E	7	10	Anno
Stipendio	E	7	10	Anno
Evento	E	200	80	Anno
Cliente	E	300	200	Anno
Abbonamento	E	300	220	Anno
Fumetto	E	250	50	Mese
Magazzino	E	500	100	Mese
Casa Editrice	E	4	10	Anno
Fornitore	E	8	12	Anno
Fornitura	E	300	200	Anno
Si iscrive	A	100	50	Mese
Acquista	A	170	40	Settimana
Prenota	A	100	30	Settimana
Fornisce	A	150	75	Mese
Spedisce	A	169	231	Mese

1.8 Vincoli d'integrità

Esistono due tipologie di vincoli: statici e dinamici. I vincoli statici sono quelli che limitano i valori assumibili da alcuni attributi indipendentemente dal tempo, mentre, sono detti vincoli dinamici quelli che riguardano valori che cambiano nel tempo. Tra questi vincoli vi sono quelli per la definizione della chiave primaria, e i vincoli che impongono regole sui valori assunti da due o più colonne.

Vincoli Statici

Questa tipologia di vincoli è rappresentata dalla definizione di chiavi primarie ed esterne

Vincoli Dinamici

- Lo stipendio di un singolo impiegato deve essere compreso tra 1100 e 1300;
- L'importo dello stipendio viene assegnato in base alla mansione.
- La data di inizio della presenza non può essere successiva a quella di fine.
- Ogni evento deve avere una capienza massima determinata.
- La data di consegna non può essere successiva a quella d'acquisto.
- L'abbonamento non deve superare la scadenza prevista dalla sua tipologia.

1.9 Verifica di normalità

Nella verifica di normalità si combatte la ridondanza attraverso la caccia alle dipendenze funzionali anomale. Mentre il rispetto della prima forma normale è in qualche modo dato per scontato, implicito nel modello relazionale, la seconda e la terza vanno verificate analizzando il significato degli attributi.

1.9.1 Prima forma normale

Una relazione è in prima forma normale se e solo se ogni componente di ogni tupla ha un valore atomico, non ulteriormente scomponibile. La prima forma normale è di norma implicita: ogni informazione deve essere “atomica”, cioè un campo deve contenere una e una sola informazione. L'attributo composto, invece, è indirizzato, scomposto in: via, CAP, città.

1.9.2 Seconda forma normale

La definizione della seconda forma normale: uno schema è in seconda forma normale se e solo se ogni attributo non primo dipende completamente dalla chiave. Se il valore di un campo può variare ad ogni variazione della chiave, possiamo dire che quel campo dipende dall'intera chiave, mentre se il valore di un altro campo può variare solo alla variazione di una parte della chiave, allora quel campo non dipende dalla chiave, ma solo da una parte di essa.

1.9.3 Terza forma normale

La terza forma normale è rispettata quando è rispettata la seconda e quando tutti gli attributi non chiave dipendono esclusivamente dalla chiave e non da altri attributi; la terza forma normale elimina la dipendenza transitiva degli attributi della chiave. La BCNF è rispettata se la terza forma normale è rispettata e se per ogni dipendenza funzionale e non banale $X \rightarrow Y$, X è una superchiave.

2. Implementazione

2.1 Creazione utenti

Il primo passo da compiere è accedere al DBMS come amministratore di sistema e creare l'utente proprietario della base di dati. È possibile creare contestualmente anche altri utenti, sebbene essi non possano ricevere alcun privilegio di oggetto, poiché lo schema ancora non esiste.

```
CREATE USER DB_starshop IDENTIFIED BY db_Admin;  
CREATE USER Cliente IDENTIFIED BY Cliente;  
CREATE USER Personale IDENTIFIED BY Staff;  
GRANT ALL PRIVILEGES TO DB_starshop;
```

In seguito, occorre disconnettersi e riconnettersi con le credenziali dell'utente DB_starshop, che è l'amministratore della nuova base di dati e che avrà i permessi per creare tutti gli oggetti che occorrono.

2.2 Data Definition Language (DDL)

Mediante il comando CREATE TABLE, vengono create le tabelle. Le istruzioni di tale comando includono tutti i vincoli di integrità esprimibili nel modello relazionale. Poiché in Oracle il tipo DATE include l'ora, i campi di tipo data e ora, pur se separati nel diagramma E/R, sono tradotti con un campo unico di tipo DATE.

2.2.1 Cliente

```
CREATE TABLE CLIENTE  
(  
    CF VARCHAR(16),  
    Nome VARCHAR(10),  
    Cognome VARCHAR(20),  
    DN DATE,  
    Telefono NUMBER(15),  
    Via VARCHAR(20),  
    Città VARCHAR(10),  
    CAP NUMBER(10),  
    codCarta NUMBER(5) NOT NULL,  
    CONSTRAINT PK_CLIENT PRIMARY KEY (CF),  
    CONSTRAINT FK_ABBONAMENT FOREIGN KEY (codCarta) REFERENCES ABBONAMENTO (codCarta)  
);
```

2.2.2 Fumetto

```
CREATE TABLE FUMETTO
(
    ISBN VARCHAR(20),
    Nome VARCHAR(20),
    Genere VARCHAR(10),
    Prezzo NUMBER(3,2),
    Num_scaffale NUMBER(2),
    codCasaEdi NUMBER(10),
    CONSTRAINT PK_FUMETTO PRIMARY KEY (ISBN),
    CONSTRAINT FK_MAGAZZIN FOREIGN KEY (Num_scaffale) REFERENCES MAGAZZINO (Num_scaffale),
    CONSTRAINT FK_CASAEDI FOREIGN KEY (codCasaEdi) REFERENCES CASA_EDITRICE (codCasaEdi)
);
```

2.2.3 Abbonamento

```
CREATE TABLE ABBONAMENTO (
codCarta NUMBER(5),
Prezzo NUMBER(5),
Durata VARCHAR(12),
Sconto NUMBER(5),
data_scadenza DATE NULL,
CONSTRAINT PK_ABBONAMENTO PRIMARY KEY (codCarta)
);
```

2.2.4 Magazzino

```
CREATE TABLE MAGAZZINO
(
    Num_scaffale NUMBER(2),
    Reparto VARCHAR(10),
    Num_piano NUMBER(2),
    CONSTRAINT PK_MAGAZZ PRIMARY KEY (Num_scaffale)
);
```

2.2.5 Casa editrice

```
CREATE TABLE CASA_EDITRICE (
    codCasaEdi NUMBER(10),
    Nome VARCHAR (15),
    Città VARCHAR (10),
    CONSTRAINT PK_CASA_EDITRICE PRIMARY KEY (codCasaEdi)
);
```

2.2.6 Fornitore

```
CREATE TABLE FORNITORE(  
  p_iva NUMBER(11),  
  nome_fornitore VARCHAR(10),  
  CONSTRAINT PK_FORNITORE PRIMARY KEY (p_iva)  
);
```

2.2.7 Fornitura

```
CREATE TABLE FORNITURA(  
  idOrdine NUMBER(5),  
  data_acquisto DATE,  
  data_consegna DATE,  
  CONSTRAINT PK_FORNITURA PRIMARY KEY (idOrdine)  
);
```

2.2.8 Personale

```
CREATE TABLE PERSONALE(  
  numTesserino NUMBER(11),  
  Nome VARCHAR(10),  
  Cognome VARCHAR(20),  
  Mansione VARCHAR(15),  
  CONSTRAINT PK_PERSONALE PRIMARY KEY (numTesserino)  
);
```

2.2.9 Turno effettuato

```
CREATE TABLE TURNO_EFFETTUATO(  
  data_oraIngressoeff DATE,  
  data_oraUscita_eff DATE,  
  numTesserino NUMBER(11),  
  CONSTRAINT PK_TURNO_PROG PRIMARY KEY (data_oraIngressoeff),  
  CONSTRAINT FK_PERSON FOREIGN KEY (numTesserino) REFERENCES PERSONALE (numTesserino)  
);
```

2.2.10 Turno programmato

```
CREATE TABLE TURNO_PROGRAMMATO(  
  data_oraIngresso DATE,  
  data_oraUscita DATE,  
  data_oraIngressoeff DATE,  
  CONSTRAINT PK_TURNO_PRO PRIMARY KEY (data_oraIngresso),  
  CONSTRAINT FK_TURNO_EFF FOREIGN KEY (data_oraIngressoeff) REFERENCES TURNO_EFFETTUATO (data_oraIngressoeff)  
);
```

2.2.11 Evento

```
CREATE TABLE EVENTO(  
  numEvento NUMBER(5),  
  Data_evento DATE,  
  Durata VARCHAR(10),  
  Tipo VARCHAR(15),  
  Capienza NUMBER(3),  
  numTesserino NUMBER(11),  
  CONSTRAINT PK_EVENT PRIMARY KEY (numEvento),  
  CONSTRAINT FK_PERS FOREIGN KEY (numTesserino) REFERENCES PERSONALE (numTesserino)  
);
```

2.2.12 Stipendio

```
CREATE TABLE STIPENDIO(  
  cod_bustapaga NUMBER(5),  
  Data_accredito DATE,  
  importo NUMBER(6),  
  numTesserino NUMBER(11),  
  CONSTRAINT PK_STIPENDIO PRIMARY KEY (cod_bustapaga),  
  CONSTRAINT FK_PERSONAL FOREIGN KEY (numTesserino) REFERENCES PERSONALE (numTesserino)  
);
```

2.2.13 Si iscrive

```
CREATE TABLE SI_ISCRIVE(  
  CF_c1 VARCHAR(16),  
  numEvento_1 NUMBER(5),  
  data_iscrizione DATE,  
  CONSTRAINT PK_ISCRIZ PRIMARY KEY (CF_c1,numEvento_1),  
  CONSTRAINT FK_CLI FOREIGN KEY (CF_c1) REFERENCES CLIENTE (CF),  
  CONSTRAINT FK_EVE FOREIGN KEY (numEvento_1) REFERENCES EVENTO (numEvento)  
);
```

2.2.14 Acquista

```
CREATE TABLE ACQUISTA(  
  CF_c2 VARCHAR(16),  
  ISBN_f1 VARCHAR(20),  
  data_acquisto DATE,  
  num_copie NUMBER(2),  
  CONSTRAINT PK_ACQUIST PRIMARY KEY (CF_c2,ISBN_f1),  
  CONSTRAINT FK_CL FOREIGN KEY (CF_c2) REFERENCES CLIENTE (CF),  
  CONSTRAINT FK_FUM FOREIGN KEY (ISBN_f1) REFERENCES FUMETTO (ISBN)  
);
```

2.2.15 Prenota

```
CREATE TABLE PRENOTA(  
  CF_c3 VARCHAR(16),  
  ISBN_f2 VARCHAR(20),  
  data_prenotazione DATE,  
  CONSTRAINT PK_ACQUIST PRIMARY KEY (CF_c3,ISBN_f2),  
  CONSTRAINT FK_CL FOREIGN KEY (CF_c3) REFERENCES CLIENTE (CF),  
  CONSTRAINT FK_FUM FOREIGN KEY (ISBN_f2) REFERENCES FUMETTO (ISBN)  
);
```

2.2.16 Fornisce

```
CREATE TABLE FORNISCE(  
  idOrdine_f1 NUMBER(5),  
  ISBN_f3 VARCHAR(20),  
  CONSTRAINT PK_FORNISCE PRIMARY KEY (idOrdine_f1,ISBN_f3),  
  CONSTRAINT FK_FORNITU FOREIGN KEY (idOrdine_f1) REFERENCES FORNITURA (idOrdine),  
  CONSTRAINT FK_FUME FOREIGN KEY (ISBN_f2) REFERENCES FUMETTO (ISBN)  
);
```

2.2.17 Spedisce

```
CREATE TABLE SPEDISCE(  
  idOrdine_f2 NUMBER(5),  
  p_iva_f1 NUMBER(11),  
  CONSTRAINT PK_SPEDISCE PRIMARY KEY (idOrdine_f2,p_iva_f1)  
  CONSTRAINT FK_FORNIT FOREIGN KEY (idOrdine_f2) REFERENCES FORNITURA (idOrdine),  
  CONSTRAINT FK_FORNI FOREIGN KEY (p_iva_f1) REFERENCES FORNITORE (p_iva)  
);
```


2.3 Data Manipulation Language (DML)

Tutte le tabelle sono popolate attraverso delle operazioni dirette di INSERT, ad eccezione della tabella Prenotazione il cui popolamento avviene attraverso l'apposita procedura, che la facilitano e garantiscono i controlli di integrità sui dati.

2.3.1 Cliente

```
INSERT INTO CLIENTE VALUES ('FNZLGU38M10M289U','Luigi','Finizio',TO_DATE('10-07-2001','DD-MM-YYYY--'),'3214265484','Via Brombeis','Napoli','80147','2465');
INSERT INTO CLIENTE VALUES ('CTNLUCU38M10E054E','Luca','Cantone',TO_DATE('30-03-2000','DD-MM-YYYY--'),'3336580264','Via Manzoni','Napoli','80147','6485');
INSERT INTO CLIENTE VALUES ('MRRMHL00C30F839L','Michele','Morrone',TO_DATE('1-01-1999','DD-MM-YYYY--'),'3342468921','Via Arpino','Casoria','80026','6247');
INSERT INTO CLIENTE VALUES ('DLCLCU03H48F839V','Lucia','De Lucia',TO_DATE('8-06-2003','DD-MM-YYYY--'),'3386542165','Via Angelis','Napoli','80147','8899');
INSERT INTO CLIENTE VALUES ('DNTNDR95E05F839L','Andrea','Di Natale',TO_DATE('5-05-1995','DD-MM-YYYY--'),'3396541257','Via Borgo Caputo','Caserta','80250','5641');
INSERT INTO CLIENTE VALUES ('DNGGTN00D17F839M','Gaetano','D'angelo',TO_DATE('17-04-2000','DD-MM-YYYY--'),'3341256871','Via Caldieri','Napoli','80147','2678');
INSERT INTO CLIENTE VALUES ('CPPLGU02P18F839N','Luigi','Coppola',TO_DATE('18-09-2002','DD-MM-YYYY--'),'331256987','Via dei mille','Napoli','80147','2534');
INSERT INTO CLIENTE VALUES ('CLNMRC01P11F839I','Marco','Calone',TO_DATE('11-09-2001','DD-MM-YYYY--'),'3324268754','Via scarlatti','Napoli','80147','3495');
INSERT INTO CLIENTE VALUES ('DRZNLG01P51F839O','Angela','Durazzi',TO_DATE('24-12-1999','DD-MM-YYYY--'),'3335456495','Via monteoliveto','Napoli','80147','6329');
INSERT INTO CLIENTE VALUES ('NRDNNA04S56F839G','Anna','Nardone',TO_DATE('16-11-2004','DD-MM-YYYY--'),'3349575624','Via foria','Napoli','80147','5654');
INSERT INTO CLIENTE VALUES ('NRDMRZ02M07F839T','Maurizio','Nardi',TO_DATE('7-08-2002','DD-MM-YYYY--'),'3337652459','Via Benevento','Casalnuovo','80038','1264');
INSERT INTO CLIENTE VALUES ('RSSRLA03H15F839G','Aurelio','Rossi',TO_DATE('15-06-1993','DD-MM-YYYY--'),'3388897445','Via bellini','Casoria','80026','2185');
INSERT INTO CLIENTE VALUES ('BNCLSS97C08F839D','Alessio','Bianchi',TO_DATE('8-03-1997','DD-MM-YYYY--'),'3665481549','Via viareggio','Arzano','80034','3694');
INSERT INTO CLIENTE VALUES ('VRDLSU03A42F839J','Luisa','Verdi',TO_DATE('2-01-2003','DD-MM-YYYY--'),'3354241697','Via chiaia','Napoli','80147','3465');
INSERT INTO CLIENTE VALUES ('PNAFDN04D09F839T','Ferdinando','Paone',TO_DATE('9-04-2004','DD-MM-YYYY--'),'3313376514','Via partenope','Napoli','80147','5642');
```

2.3.2 Fumetto

```
INSERT INTO FUMETTO VALUES ('978-5-8372-5781-0','One Piece','shonen','5','2','0030');
INSERT INTO FUMETTO VALUES ('978-3-7224-3230-4','Tokyo ghoul','shonen','5','3','2530');
INSERT INTO FUMETTO VALUES ('978-6-1127-8750-3','Chainsaw Man','shonen','5','4','1120');
INSERT INTO FUMETTO VALUES ('978-0-4038-5065-5','Pokemon','Kodomo','4','5','2530');
INSERT INTO FUMETTO VALUES ('978-0-4995-5693-6','Attack on titan','shonen','6','6','1120');
INSERT INTO FUMETTO VALUES ('978-6-4369-5128-4','Naruto','shonen','5','7','1154');
INSERT INTO FUMETTO VALUES ('978-1-8428-9495-8','Bleach','shonen','5','8','1154');
INSERT INTO FUMETTO VALUES ('978-8-0092-9106-3','Jujutsu Kaisen','shonen','4.50','9','1154');
INSERT INTO FUMETTO VALUES ('978-3-6841-0819-4','Tokyo Revengers','shonen','5','10','2530');
INSERT INTO FUMETTO VALUES ('978-3-0243-2220-4','Hunter x Hunter','seinen','8','11','1154');
INSERT INTO FUMETTO VALUES ('978-0-8902-8032-4','Demon Slayer','shonen','5','12','0030');
INSERT INTO FUMETTO VALUES ('978-1-8446-4545-9','Banana Fish','josei','9','13','1154');
INSERT INTO FUMETTO VALUES ('978-4-1594-9625-9','Calata Capodichino','Fumetto','4','14','3322');
```

2.3.3 Abbonamento

```
INSERT INTO ABBONAMENTO VALUES ('2465','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('6485','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('6247','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('8899','2','Mensile','5','');
INSERT INTO ABBONAMENTO VALUES ('5641','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('2678','4','Trimestrale','5','');
INSERT INTO ABBONAMENTO VALUES ('2534','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('3495','2','Mensile','5','');
INSERT INTO ABBONAMENTO VALUES ('6329','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('5654','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('1264','2','Mensile','5','');
INSERT INTO ABBONAMENTO VALUES ('2185','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('3694','4','Trimestrale','5','');
INSERT INTO ABBONAMENTO VALUES ('3465','6','Annuale','5','');
INSERT INTO ABBONAMENTO VALUES ('5642','2','Mensile','5','');
```

2.3.4 Magazzino

```
INSERT INTO MAGAZZINO VALUES ('1','shonen','1');
INSERT INTO MAGAZZINO VALUES ('2','shonen','1');
INSERT INTO MAGAZZINO VALUES ('3','shonen','1');
INSERT INTO MAGAZZINO VALUES ('4','shonen','1');
INSERT INTO MAGAZZINO VALUES ('5','Kodomo','1');
INSERT INTO MAGAZZINO VALUES ('6','shonen','2');
INSERT INTO MAGAZZINO VALUES ('7','shonen','2');
INSERT INTO MAGAZZINO VALUES ('8','shonen','2');
INSERT INTO MAGAZZINO VALUES ('9','shonen','2');
INSERT INTO MAGAZZINO VALUES ('10','shonen','2');
INSERT INTO MAGAZZINO VALUES ('11','seinen','3');
INSERT INTO MAGAZZINO VALUES ('12','shonen','3');
INSERT INTO MAGAZZINO VALUES ('13','josei','3');
INSERT INTO MAGAZZINO VALUES ('14','fumetto','3');
INSERT INTO MAGAZZINO VALUES ('15','shonen','3');
```

2.3.5 Casa editrice

```
INSERT INTO CASA_EDITRICE VALUES ('0030','StarComics','Perugia');
INSERT INTO CASA_EDITRICE VALUES ('1120','PlanetManga','Modena');
INSERT INTO CASA_EDITRICE VALUES ('2530','J-Pop','Milano');
INSERT INTO CASA_EDITRICE VALUES ('1154','Panini','Perugia');
INSERT INTO CASA_EDITRICE VALUES ('3322','Shockdom','Perugia');
```

2.3.6 Fornitore

```
INSERT INTO FORNITORE VALUES ('48887547306','MangaShop');
INSERT INTO FORNITORE VALUES ('65408950756','Manicomix');
INSERT INTO FORNITORE VALUES ('65408950756','StarStore');
INSERT INTO FORNITORE VALUES ('17433003880','AnkorStore');
INSERT INTO FORNITORE VALUES ('35144637066','ComicsGroup');
INSERT INTO FORNITORE VALUES ('18450935494','SoloStocks');
```

2.3.7 Fornitura

```
INSERT INTO FORNITURA VALUES ('26269',TO_DATE('1-06-2022','DD-MM-YYYY'),TO_DATE('08-06-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('46059',TO_DATE('1-06-2022','DD-MM-YYYY'),TO_DATE('08-06-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('37105',TO_DATE('1-06-2022','DD-MM-YYYY'),TO_DATE('10-06-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('73837',TO_DATE('15-06-2022','DD-MM-YYYY'),TO_DATE('24-06-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('94592',TO_DATE('2-07-2022','DD-MM-YYYY'),TO_DATE('9-07-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('39134',TO_DATE('2-07-2022','DD-MM-YYYY'),TO_DATE('8-07-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('49965',TO_DATE('20-07-2022','DD-MM-YYYY'),TO_DATE('27-07-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('33397',TO_DATE('20-07-2022','DD-MM-YYYY'),TO_DATE('28-07-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('28922',TO_DATE('10-09-2022','DD-MM-YYYY'),TO_DATE('16-09-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('56589',TO_DATE('10-09-2022','DD-MM-YYYY'),TO_DATE('19-09-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('77099',TO_DATE('16-09-2022','DD-MM-YYYY'),TO_DATE('26-09-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('43988',TO_DATE('16-09-2022','DD-MM-YYYY'),TO_DATE('24-09-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('90342',TO_DATE('18-09-2022','DD-MM-YYYY'),TO_DATE('22-09-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('88933',TO_DATE('1-10-2022','DD-MM-YYYY'),TO_DATE('8-10-2022','DD-MM-YYYY'));
INSERT INTO FORNITURA VALUES ('51828',TO_DATE('1-10-2022','DD-MM-YYYY'),TO_DATE('8-10-2022','DD-MM-YYYY'));
```

2.3.8 Personale

```
INSERT INTO PERSONALE VALUES ('754','Andrea','Mazza','Scaffalista');
INSERT INTO PERSONALE VALUES ('123','Angelo','Farina','Magazziniere');
INSERT INTO PERSONALE VALUES ('321','Vincenzo','Lupino','Cassiere');
INSERT INTO PERSONALE VALUES ('156','Luisa','Marino','Scaffalista');
INSERT INTO PERSONALE VALUES ('257','Angela','Fontana','Scaffalista');
INSERT INTO PERSONALE VALUES ('357','Maurizio','Ferrari','Cassiere');
INSERT INTO PERSONALE VALUES ('259','Antonio','Russo','Magazziniere');
INSERT INTO PERSONALE VALUES ('741','Antonella','Costa','Magazziniere');
INSERT INTO PERSONALE VALUES ('654','Angela','Gallo','Scaffalista');
```

2.3.9 Turno effettuato

```
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('18-05-2022 8:10','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 12:55','DD-MM-YYYY HH24:MI'),'745');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('18-05-2022 14:10','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 20:00','DD-MM-YYYY HH24:MI'),'754');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('19-05-2022 8:20','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 13:00','DD-MM-YYYY HH24:MI'),'123');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('19-05-2022 14:05','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 19:40','DD-MM-YYYY HH24:MI'),'321');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('20-05-2022 8:13','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 12:55','DD-MM-YYYY HH24:MI'),'156');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('20-05-2022 14:07','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 19:43','DD-MM-YYYY HH24:MI'),'257');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('21-05-2022 8:02','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 12:50','DD-MM-YYYY HH24:MI'),'357');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('21-05-2022 14:30','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 20:00','DD-MM-YYYY HH24:MI'),'259');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('22-05-2022 8:04','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 12:48','DD-MM-YYYY HH24:MI'),'741');
INSERT INTO TURNO_EFFETTUATO VALUES(TO_DATE('22-05-2022 14:08','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 19:34','DD-MM-YYYY HH24:MI'),'654');
```

2.3.10 Turno programmato

```
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('18-05-2022 8:00','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 13:00','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 8:10','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('18-05-2022 14:00','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 20:00','DD-MM-YYYY HH24:MI'),TO_DATE('18-05-2022 14:10','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('19-05-2022 8:00','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 13:00','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 8:20','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('19-05-2022 14:00','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 20:00','DD-MM-YYYY HH24:MI'),TO_DATE('19-05-2022 14:05','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('20-05-2022 8:00','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 13:00','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 8:13','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('20-05-2022 14:00','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 20:00','DD-MM-YYYY HH24:MI'),TO_DATE('20-05-2022 14:07','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('21-05-2022 8:00','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 13:00','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 8:02','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('21-05-2022 14:00','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 20:00','DD-MM-YYYY HH24:MI'),TO_DATE('21-05-2022 14:30','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('22-05-2022 8:00','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 13:00','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 8:04','DD-MM-YYYY HH24:MI'));
INSERT INTO TURNO_PROGRAMMATO VALUES(TO_DATE('22-05-2022 14:00','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 20:00','DD-MM-YYYY HH24:MI'),TO_DATE('22-05-2022 14:08','DD-MM-YYYY HH24:MI'));
```


2.3.11 Evento

```
INSERT INTO EVENTO VALUES ('1',TO_DATE('01-06-2022','DD-MM-YYYY--'),'10 giorni','Firmacopie','75','745');
INSERT INTO EVENTO VALUES ('2',TO_DATE('11-06-2022','DD-MM-YYYY--'),'10 giorni','Presentazione','75','754');
INSERT INTO EVENTO VALUES ('3',TO_DATE('21-06-2022','DD-MM-YYYY--'),'7 giorni','Firmacopie','75','123');
INSERT INTO EVENTO VALUES ('4',TO_DATE('29-06-2022','DD-MM-YYYY--'),'7 giorni','Firmacopie','75','321');
INSERT INTO EVENTO VALUES ('5',TO_DATE('01-05-2022','DD-MM-YYYY--'),'7 giorni','Presentazione','75','156');
INSERT INTO EVENTO VALUES ('6',TO_DATE('08-05-2022','DD-MM-YYYY--'),'10 giorni','Firmacopie','75','257');
INSERT INTO EVENTO VALUES ('7',TO_DATE('15-05-2022','DD-MM-YYYY--'),'15 giorni','Presentazione','75','357');
INSERT INTO EVENTO VALUES ('8',TO_DATE('15-05-2022','DD-MM-YYYY--'),'7 giorni','Firmacopie','75','259');
INSERT INTO EVENTO VALUES ('9',TO_DATE('15-05-2022','DD-MM-YYYY--'),'15 giorni','Presentazione','75','741');
INSERT INTO EVENTO VALUES ('10',TO_DATE('1-07-2022','DD-MM-YYYY--'),'10 giorni','Firmacopie','75','654');
INSERT INTO EVENTO VALUES ('11',TO_DATE('5-07-2022','DD-MM-YYYY--'),'15 giorni','Presentazione','75','745');
INSERT INTO EVENTO VALUES ('12',TO_DATE('15-07-2022','DD-MM-YYYY--'),'15 giorni','Presentazione','75','357');
INSERT INTO EVENTO VALUES ('13',TO_DATE('30-07-2022','DD-MM-YYYY--'),'10 giorni','Presentazione','75','259');
INSERT INTO EVENTO VALUES ('14',TO_DATE('5-08-2022','DD-MM-YYYY--'),'7 giorni','Firmacopie','75','741');
INSERT INTO EVENTO VALUES ('15',TO_DATE('15-08-2022','DD-MM-YYYY--'),'10 giorni','Firmacopie','75','321');
```

2.3.12 Stipendio

```
INSERT INTO STIPENDIO VALUES ('14242',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1200','745');
INSERT INTO STIPENDIO VALUES ('15912',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1300','754');
INSERT INTO STIPENDIO VALUES ('45647',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1200','123');
INSERT INTO STIPENDIO VALUES ('32416',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1100','321');
INSERT INTO STIPENDIO VALUES ('12743',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1300','156');
INSERT INTO STIPENDIO VALUES ('74325',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1300','257');
INSERT INTO STIPENDIO VALUES ('46211',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1100','357');
INSERT INTO STIPENDIO VALUES ('01125',TO_DATE('13-06-2022','DD-MM-YYYY--'),'1200','259');
INSERT INTO STIPENDIO VALUES ('23654',TO_DATE('15-06-2022','DD-MM-YYYY--'),'1200','741');
INSERT INTO STIPENDIO VALUES ('12465',TO_DATE('15-06-2022','DD-MM-YYYY--'),'1300','654');
```

2.3.13 Si iscrive

```
INSERT INTO SI_ISCRIVE VALUES ('FNZLGU38M10M289U','1',TO_DATE('30-05-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('CTNLUCU38M10E054E','2',TO_DATE('1-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('MRRMHL00C30F839L','3',TO_DATE('10-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('DLCLCU03H48F839V','4',TO_DATE('20-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('DNTNDR95E05F839L','5',TO_DATE('28-04-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('DNGGTN00D17F839M','6',TO_DATE('30-04-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('CPPLGU02P18F839N','7',TO_DATE('10-05-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('CLNMRC01P11F839I','8',TO_DATE('09-05-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('DRZNGLO1P51F839O','9',TO_DATE('04-05-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('NRDNNA04S56F839G','10',TO_DATE('30-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('NRDMRZ02M07F839T','11',TO_DATE('29-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('RSSRLA93H15F839G','12',TO_DATE('24-06-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('BNCLSS97C08F839D','13',TO_DATE('10-07-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('VRDLSU03A42F839J','14',TO_DATE('27-07-2022','DD-MM-YYYY'));
INSERT INTO SI_ISCRIVE VALUES ('PNAFDN04D09F839T','15',TO_DATE('1-08-2022','DD-MM-YYYY'));
```

2.3.14 Acquista

```
INSERT INTO ACQUISTA VALUES ('FNZLGU38M10M289U','978-7-5706-6410-8',TO_DATE('15-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('CTNLUCU38M10E054E','978-5-8372-5781-0',TO_DATE('17-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('MRRMHL00C30F839L','978-3-7224-3230-4',TO_DATE('17-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('DLCLCU03H48F839V','978-6-1127-8750-3',TO_DATE('10-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('DNTNDR95E05F839L','978-0-4038-5065-5',TO_DATE('16-06-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('DNGGTN00D17F839M','978-0-4995-5693-6',TO_DATE('17-06-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('CPPLGU02P18F839N','978-6-4369-5128-4',TO_DATE('10-06-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('CLNMRC01P11F839I','978-1-8428-9495-8',TO_DATE('5-06-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('DRZNGLO1P51F839O','978-8-0092-9106-3',TO_DATE('16-07-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('NRDNNA04S56F839G','978-3-6841-0819-4',TO_DATE('9-06-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('NRDMRZ02M07F839T','978-3-0243-2220-4',TO_DATE('10-04-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('RSSRLA93H15F839G','978-0-8902-8032-4',TO_DATE('6-08-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('BNCLSS97C08F839D','978-1-8446-4545-9',TO_DATE('10-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('VRDLSU03A42F839J','978-4-1594-9625-9',TO_DATE('12-05-2022','DD-MM-YYYY--'));
INSERT INTO ACQUISTA VALUES ('PNAFDN04D09F839T','978-3-5496-9976-8',TO_DATE('2-06-2022','DD-MM-YYYY--'));
```

2.3.15 Prenota

```
INSERT INTO PRENOTA VALUES ('FNZLGU38M10M289U','978-7-5706-6410-8',TO_DATE('15-05-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('CTNLUCU38M10E054E','978-5-8372-5781-0',TO_DATE('14-05-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('MRRMHL00C30F839L','978-3-7224-3230-4',TO_DATE('19-05-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('DLCLCU03H48F839V','978-6-1127-8750-3',TO_DATE('29-05-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('DNTNDR95E05F839L','978-0-4038-5065-5',TO_DATE('3-05-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('DNGGTN00D17F839M','978-0-4995-5693-6',TO_DATE('4-06-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('CPPLGU02P18F839N','978-6-4369-5128-4',TO_DATE('8-06-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('CLNMRC01P11F839I','978-1-8428-9495-8',TO_DATE('30-06-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('DRZNGLO1P51F839O','978-8-0092-9106-3',TO_DATE('1-06-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('NRDNNA04S56F839G','978-3-6841-0819-4',TO_DATE('6-06-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('NRDMRZ02M07F839T','978-3-0243-2220-4',TO_DATE('1-07-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('RSSRLA93H15F839G','978-0-8902-8032-4',TO_DATE('10-07-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('BNCLSS97C08F839D','978-1-8446-4545-9',TO_DATE('4-07-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('VRDLSU03A42F839J','978-4-1594-9625-9',TO_DATE('14-08-2022','DD-MM-YYYY--'));
INSERT INTO PRENOTA VALUES ('PNAFDN04D09F839T','978-3-5496-9976-8',TO_DATE('1-08-2022','DD-MM-YYYY--'));
```

2.3.16 Fornisce

```
INSERT INTO FORNISCE VALUES ('26269','978-7-5706-6410-8');
INSERT INTO FORNISCE VALUES ('46059','978-5-8372-5781-0');
INSERT INTO FORNISCE VALUES ('37105','978-3-7224-3230-4');
INSERT INTO FORNISCE VALUES ('73837','978-6-1127-8750-3');
INSERT INTO FORNISCE VALUES ('94592','978-0-4038-5065-5');
INSERT INTO FORNISCE VALUES ('39134','978-0-4995-5693-6');
INSERT INTO FORNISCE VALUES ('49965','978-6-4369-5128-4');
INSERT INTO FORNISCE VALUES ('33397','978-1-8428-9495-8');
INSERT INTO FORNISCE VALUES ('28922','978-8-0092-9106-3');
INSERT INTO FORNISCE VALUES ('56589','978-3-6841-0819-4');
INSERT INTO FORNISCE VALUES ('77099','978-3-0243-2220-4');
INSERT INTO FORNISCE VALUES ('43988','978-0-8902-8032-4');
INSERT INTO FORNISCE VALUES ('90342','978-1-8446-4545-9');
INSERT INTO FORNISCE VALUES ('88933','978-4-1594-9625-9');
INSERT INTO FORNISCE VALUES ('51828','978-3-5496-9976-8');
```

2.3.17 Spedisce

```
INSERT INTO SPEDISCE VALUES ('26269','48887547306');
INSERT INTO SPEDISCE VALUES ('46059','65408950756');
INSERT INTO SPEDISCE VALUES ('37105','48887547306');
INSERT INTO SPEDISCE VALUES ('73837','65408950756');
INSERT INTO SPEDISCE VALUES ('94592','48887547306');
INSERT INTO SPEDISCE VALUES ('39134','65408950756');
INSERT INTO SPEDISCE VALUES ('49965','48887547306');
INSERT INTO SPEDISCE VALUES ('33397','65408950756');
INSERT INTO SPEDISCE VALUES ('28922','35144637066');
INSERT INTO SPEDISCE VALUES ('56589','65408950756');
INSERT INTO SPEDISCE VALUES ('77099','35144637066');
INSERT INTO SPEDISCE VALUES ('43988','18450935494');
INSERT INTO SPEDISCE VALUES ('90342','18450935494');
INSERT INTO SPEDISCE VALUES ('88933','17433003880');
INSERT INTO SPEDISCE VALUES ('51828','17433003880');
```

3.3 Trigger

I trigger DML sono utili per il controllo di vincoli dinamici in fase di immissione o aggiornamento dei dati. È possibile usarli anche per calcolare, generare o sovrascrivere il valore di alcuni campi.

3.3.1 Salario Personale

Il trigger si occupa di verificare che l'importo dello stipendio deve essere compreso nel range 1100-1300.

```
CREATE OR REPLACE TRIGGER SALARIO_PERSONALE
BEFORE INSERT OR UPDATE ON STIPENDIO
FOR EACH ROW
DECLARE
IMPORTO_INVALIDO EXCEPTION;
BEGIN
IF (:NEW.importo < 1100 OR :NEW.importo > 1300) THEN
RAISE IMPORTO_INVALIDO;
END IF;
EXCEPTION
WHEN IMPORTO_INVALIDO THEN
RAISE_APPLICATION_ERROR(-20014, 'IMPORTO NON VALIDO');
END;
```

3.3.2 Variazione Stipendio

Il trigger si occupa di aggiornare il nuovo importo inserito all'interno della tabella stipendio

```
CREATE TRIGGER VariazioneStipendio
AFTER UPDATE ON STIPENDIO
FOR EACH ROW
BEGIN
    IF (:NEW.importo < :OLD.importo) THEN
        UPDATE STIPENDIO
        SET importo = :OLD.importo
        WHERE idStipendio = :NEW.idStipendio;
    END IF;
END;
```


3.3.3 Controllo data turno

Il trigger verifica che la data e l'ora delle presenze siano antecedenti a quelle di fine. Nel caso in cui, non dovesse verificarsi la condizione, il trigger si attiva rilasciando in output un codice di errore.

```
CREATE OR REPLACE TRIGGER Controllo_dataTurno
BEFORE INSERT OR UPDATE ON TURNO_EFFETTUATO
FOR EACH ROW
DECLARE
GiornoSbagliato EXCEPTION;
BEGIN
IF (:NEW.data_oraIngressoeff > :NEW.data_oraUscita_eff) THEN
RAISE controllosbagliato;
END IF;
EXCEPTION WHEN
GiornoSbagliato THEN
RAISE_APPLICATION_ERROR (-20011,'DATA DEL TURNO ERRATA, RICONTROLLARE');
END;
```

3.3.4 Capienza evento

Il trigger si occupa di aggiornare la capienza massima del negozio in vista di un evento

```
CREATE OR REPLACE TRIGGER CapienzaEvento
AFTER UPDATE ON EVENTO
FOR EACH ROW
DECLARE
CapienzaUguale EXCEPTION;
BEGIN
IF (:NEW.Capienza > 75) THEN
RAISE CapienzaUguale;
ELSE
UPDATE EVENTO
SET Capienza = :OLD.Capienza
WHERE numEvento = :NEW.numEvento;
END IF;
EXCEPTION WHEN
CapienzaUguale THEN
RAISE_APPLICATION_ERROR (-20031,'IMPOSSIBILE INSERIRE, CAPIENZA MASSIMA RAGGIUNTA');
END;
```

3.3.5 Controllo Fornitura

Il trigger si occupa di verificare che la data dell'ordine sia antecedente a quella di acquisto. Nel caso in cui, ciò non dovesse verificarsi la condizione, il trigger si attiva rilasciando in output un codice di errore.

```
CREATE OR REPLACE TRIGGER ControlloFornitura
BEFORE INSERT OR UPDATE ON FORNITURA
FOR EACH ROW
DECLARE
dataErrata EXCEPTION;
BEGIN
IF (:new.data_consegna < :NEW.data_acquisto) THEN
RAISE dataErrata;
END IF;
EXCEPTION WHEN
dataErrata THEN
RAISE_APPLICATION_ERROR (-20006, 'Errore nella consegna');
END
```

3.3.6 Data scadenza

Il trigger si occupa di calcolare la data di scadenza della carta in base al piano d'abbonamento sottoscritto dal cliente.

```
CREATE OR REPLACE TRIGGER datascadenza
BEFORE INSERT ON ABBONAMENTO
FOR EACH ROW
BEGIN
IF (:NEW.data_scadenza is NULL) THEN
IF (:NEW.Durata = 'Annuale') THEN
:NEW.data_scadenza := TO_DATE(sysdate+365);
ELSIF (:NEW.Durata = 'Trimestrale') THEN
:NEW.data_scadenza := TO_DATE(sysdate+90);
ELSIF (:NEW.Durata = 'Mensile') THEN
:NEW.data_scadenza := TO_DATE(sysdate+30);
END IF;
END IF;
END;
```

3.4 Procedure

3.4.1 Tipo salario

La procedura stampa la mansione assegnata in base all'importo dato in input

```
CREATE OR REPLACE PROCEDURE tipo_salario (numtessera NUMBER , importo NUMBER)
IS
ruolo PERSONALE %ROWTYPE;
stipendio_errato EXCEPTION; -- dichiarazione
BEGIN
    IF(importo = 1100) THEN
        DBMS_OUTPUT.PUT_LINE('Mansione assegnata : Cassiere');
    ELSIF (importo = 1200) THEN
        DBMS_OUTPUT.PUT_LINE('Mansione assegnata : Magazziniere');
    ELSIF (importo = 1300) THEN
        DBMS_OUTPUT.PUT_LINE('Mansione assegnata : Scaffalista');
    ELSIF (importo < 1100 OR importo > 1300) THEN
        RAISE stipendio_errato;
    END IF;

    SELECT * INTO ruolo
    FROM PERSONALE
    WHERE numTesserino = numTessera;
    DBMS_OUTPUT.PUT_LINE('Nome e cognome impiegato: ');
    DBMS_OUTPUT.PUT_LINE( ruolo.Nome || ' ' || ruolo.Cognome || ' ');

    EXCEPTION
    WHEN stipendio_errato THEN
        RAISE_APPLICATION_ERROR(-20002,'STIPENDIO ERRATO');
END tipo_salario;
```

3.4.2 Stampa contatti

La procedura stampa, partendo dal codice della carta passato in input, tutti i dati inerenti al cliente e al suo rispettivo abbonamento sottoscritto

```
CREATE OR REPLACE PROCEDURE stampa_contatt( numercodCarta IN NUMBER)
IS
    stamp_contatto CLIENTE %ROWTYPE;
    tipo ABBONAMENTO %ROWTYPE;
BEGIN
    SELECT * INTO stamp_contatto
    FROM CLIENTE
    WHERE codCarta = numercodCarta;

    SELECT * into tipo
    FROM ABBONAMENTO;
    DBMS_OUTPUT.PUT_LINE('Durata abbonamento: ');
    DBMS_OUTPUT.PUT_LINE( tipo.Durata || '');

    DBMS_OUTPUT.PUT_LINE('Nome e cognome cliente: ');
    DBMS_OUTPUT.PUT_LINE( stamp_contatto.Nome || ' ' || stamp_contatto.Cognome || ' ');
    EXCEPTION
    WHEN no_data_found THEN RAISE_APPLICATION_ERROR(-20010,'NESSUN UTENTE ASSOCIATO A QUESTO CODICE ');
END;
```

3.4.3 Effettua sconto

La procedura si occupa di controllare se il cliente sia in possesso di abbonamento, in tal caso viene applicato uno sconto del 5% sul prezzo del fumetto acquistato.

```
CREATE OR REPLACE PROCEDURE effettua_sconto (numCarta number , ISBN1 char )
IS
numero_tessera number;
BEGIN
SELECT codCarta INTO numero_tessera
FROM ABBONAMENTO
WHERE codCarta = numCarta;
IF (numero_tessera IS NOT NULL) THEN
UPDATE FUMETTO
SET Prezzo = Prezzo - (Prezzo * 0.05)
WHERE ISBN = ISBN1;
ELSE
WHEN no_data_found THEN RAISE_APPLICATION_ERROR(-20011,'PAGARE PREZZO PIENO IN QUANTO CLIENTE SPROVVISTO DI CARTA');
END IF;
END;
```

3.4.4 Cambio mansione

La procedura si occupa di aggiornare le tabelle inerenti allo stipendio e ai dati dell'impiegato in caso di cambio di mansione

```
CREATE OR REPLACE PROCEDURE cambio_mansione (codicetess number, ruolo varchar)
IS
numerotess NUMBER;
BEGIN

SELECT numTesserino INTO numerotess
FROM PERSONALE
WHERE numTesserino = codicetess;
IF (numerotess IS NOT NULL)then
UPDATE PERSONALE
SET Mansione = ruolo
WHERE numTesserino = codicetess;

IF (ruolo = 'Magazziniere') THEN
UPDATE STIPENDIO
SET Importo = 1300
WHERE numTesserino = codicetess;

ELSIF (ruolo = 'Cassiere') THEN
UPDATE STIPENDIO
SET Importo = 1100
WHERE numTesserino = codicetess;

ELSIF (ruolo = 'Scaffalista') THEN
UPDATE STIPENDIO
SET Importo = 1200
WHERE numTesserino = codicetess;
END IF;
END IF;
END;
```