



Relazione Basi di Dati E Laboratorio di Basi di Dati

PROGETTO NEGOZIO DI MUSICA

A CURA DI:

FULVIO SERAO 0124/002423

LUCA PEZZOLLA 0124/002411

SIMONE MICILLO 0124/002439

Sintesi dei requisiti:

Il database simula la gestione di una catena di negozi di musica. Per far ciò è necessario tenere traccia della merce che arriva in uno dei negozi, fornita dai fornitori. Un cliente ha la possibilità di acquistare uno o più prodotti in un negozio sia recandovisi fisicamente o, in alternativa, effettuando un acquisto online, facendosi spedire il contenuto del suo carrello all'indirizzo di spedizione fornito (a patto che sia tesserato). Il prezzo di spedizione è pari a 5 euro per una somma di denaro spesa inferiore ai 100 euro, altrimenti sarà gratuita. I dipendenti del negozio lavorano in un intervallo di tempo che va dalla singola ora alle dodici ore per uno stipendio che arriva fino a 1899 euro. Ogni negozio è gestito da un dirigente la cui paga è ovviamente superiore a quella di un semplice dipendente.

Glossario:

Termine	Definizione	Sinonimi
Cliente	Persona deputata all'acquisto di prodotti in uno dei negozi	Acquirente, compratore
Impiegato	Persona che lavora in uno dei negozi	Dipendente, addetto
Manager	Persona deputata alla gestione di un negozio	Principale, titolare
Spedizionario	Persona deputata a spedire la merce acquistata all'indirizzo fornito dal cliente	Corriere, trasportatore
Magazziniere	Persona deputata al controllo della merce in negozio	Custode, sorvegliante

Diagramma EE/R:

Il diagramma EE/R è visibile nella figura sottostante

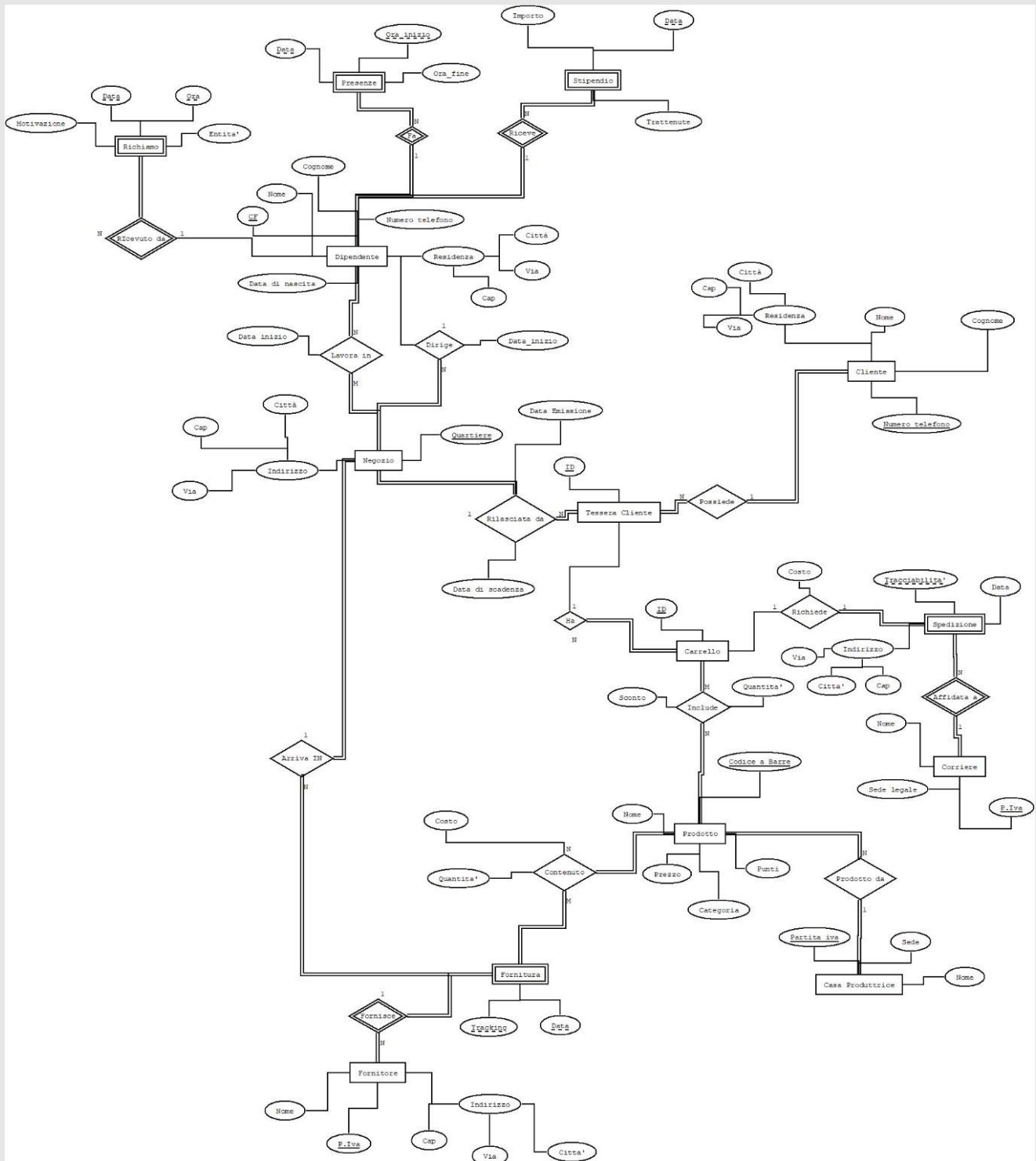
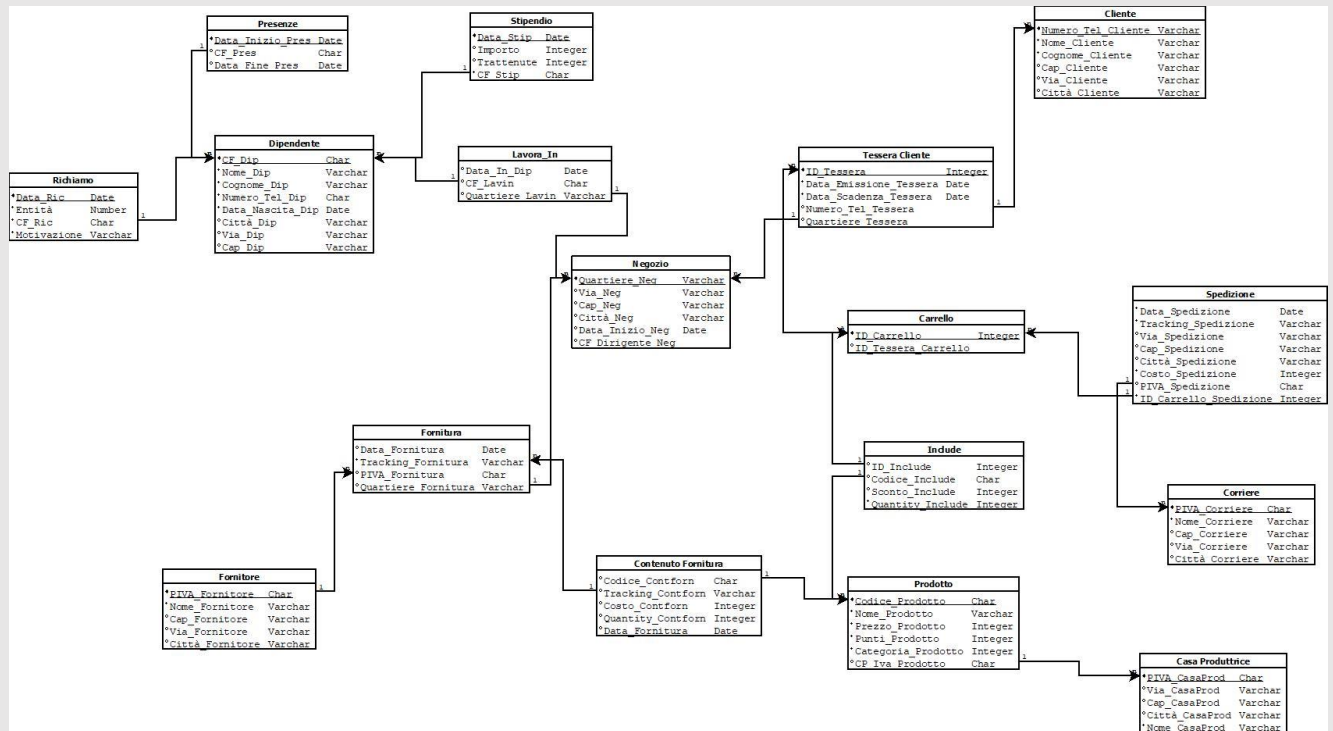


Diagramma relazionale:

Il diagramma relazionale è visibile nella figura sottostante



Utenti e loro categorie:

Utente	Tipo	Volume	Permessi
Cliente	Comune	1	Execute Acquista e Select on Prodotto
Magazziniere	Comune	1	Execute Rifornimento e Select, Insert, Update on Fornitura, Contenuto_Forn e Select on Fornitore
Manager	Comune	1	Execute Dip_Modello, Dip_Pessimo, Scontistica e Select, Insert, Update on Richiamo, Stipendio, Dipendente, Presenze, Lavora_In, Negozio, Prodotto, Casa_Prod, Fornitore, Cliente, Tessera Select on Fornitura , Corriere, Spedizione, Contenuto_Forn
Spedizioniere	Comune	1	Execute Sped_Tex e Select, Insert, Update on Spedizione, Carrello, Corriere
Tecnico	Amministratore	1	ALL

Operazione degli utenti:

Le operazioni accessibili agli utenti sono sei: Acquista, Dip_Modello, Dip_Pessimo, Rifornimento, Sped_Tex e Scontistica.

1. *Acquista*

Scopo: Permettere l'acquisto di un prodotto ad un cliente.

Argomenti: Numero di tessera, codice prodotto, quantità, ID Carrello, Sconto

Risultati: Aggiunta prodotto nel carrello

Usa: Carrello, Include, Prodotto, Tessera

Modifica: Carrello, Prodotto, Include

2. *Dip_Modello*

Scopo: Concede un bonus stipendio di mille euro se il dipendente lavora più di tutti e non ha mai ricevuto un richiamo

Argomenti: Negozio di un determinato quartiere

Risultati: Aggiunge mille euro al prossimo stipendio del dipendente

Usa: Dipendente, Lavora_In, Negozio, Stipendio

Modifica: Stipendio

3. *Dip_Pessimo*

Scopo: Diminuisce lo stipendio di 300 euro al dipendente che ha ricevuto almeno un richiamo e lavora meno ore di tutti gli altri (si applica la diminuzione allo stipendio precedente); aggiunge inoltre un richiamo di entità 1 con la causale 'Low_Hours'

Argomenti: Negozio di un determinato quartiere

Risultati: Diminuisce lo stipendio di 300 euro

Usa: Dipendente, Lavora_In, Negozio, Stipendio

Modifica: Stipendio

4. *Rifornimento*

Scopo: Rifornisce un negozio di prodotti

Argomenti: Tracking, Data, Prodotto, Quantità, Prezzo, Quartiere

Risultati: Rifornisce il negozio il cui quartiere è passato per argomento

Usa: Fornitura, Contenuto_Forn, Prodotto

Modifica: Contenuto_Forn

5. *Sped_Tex*

Scopo: Inserisce spedizione ad un cliente tesserato che decide di acquistare un prodotto online

Argomenti: ID Tessera, Data, Via, Cap, Città, Partita Iva, Tracking

Risultati: Inserisce spedizione ad un cliente che ne fa richiesta

Usa: Prodotto, Carrello, Tessera, Include, Spedizione

Modifica: Spedizione

6. Scontistica

Scopo: Sconta il prodotto meno venduto tra tutti i negozi della catena

Argomenti: Nessuno

Risultati: Inserisce uno sconto al prodotto meno venduto

Usa: Include, Carrello, Tessera, Prodotto

Modifica: Prodotto

Operazione	Volume	Periodo	Incremento
Richiamo	5 * n. negozi	Anno	10*n. negozi
Presenze	5 * n. dipendenti	Settimana	5 * n. dipendenti
Dipendente	15 * n. negozi	Anno	15
Stipendio	1 * n. dipendenti	Mese	1 * n. dipendenti
Lavora_In	15 * n. negozi	Anno	15
Negozi	15	Anno	1
Tessera	1000	Anno	1000
Carrello	10000 * n. negozi	Anno	20000 * n. negozi
Cliente	1000	Anno	1000
Spedizione	200	Anno	200
Corriere	10	Anno	0
Include	2 * n. carrelli	Anno	2 * n. carrelli
Prodotto	1000	Anno	100
Contenuto F.	700	Trimestrale	200
Fornitura	15	Mese	15
Fornitore	20	Anno	0
Casa Prod.	20	Anno	0

Vincoli di integrità:

Sono di seguito riportati i vincoli di integrità statici del Database:

- Un dipendente può ricevere un richiamo per aver lavorato poche ore, per aver fatto ritardo oppure per comportamento inappropriato.
- L'entità del richiamo può andare da 1 a 4, per i motivi sopracitati.
- L'ID di un carrello deve essere un intero positivo maggiore di 0.
- Un prodotto può appartenere a 4 categorie diverse (CD, Impianto Stereo, Strumento Musicale, Audio Input Devices).

Sono di seguito riportati i vincoli di integrità dinamici del Database:

- Un prodotto non può essere venduto in una quantità maggiore rispetto a quella presente in negozio.
- In un negozio non possono lavorare contemporaneamente più di 15 dipendenti (questo è implementato con un trigger, ma si assume che quest'ultimo venga inserito dopo aver immesso nella base di dati tutti i negozi della catena).
- Un dipendente non può essere eletto a manager se ha lavorato in negozio per meno di un anno.
- Un dipendente non può lavorare più di undici ore e cinquantanove minuti al giorno.
- Un dipendente non può ricevere più di nove richiami, pena licenziamento.
- Un dipendente non può lavorare in due negozi.
- Un dirigente deve ricevere uno stipendio superiore ai 3200 euro.
- Un dipendente non può ricevere uno stipendio superiore ai 1900 euro.
- Un cliente non può ricevere due tessere nello stesso anno.
- Un dipendente non può avere meno di diciotto anni o più di settanta anni.

Normalizzazione:

La verifica di normalizzazione è un processo necessario affinché si possa capire se, e in quale quantità, sia presente o meno ridondanza nello schema relazionale. E' possibile fare ciò, andando a sviscerare entità per entità tutti gli attributi, al fine di cogliere eventuali dipendenze funzionali anomali.

Il nostro schema, in particolare, dal momento che tutti gli attributi di tutte le tabelle sono atomici (come più volte chiarito durante il corso), è sicuramente in prima forma normale; tenendo poi conto che le relazioni M a N non hanno dipendenze anomale, e considerando che non v'è ridondanza in nessun attributo di nessuna tabella, ne vien facile affermare che lo schema si trova nella terza forma normale e, in particolare, in quella di Boyce e Codd.

Ovviamente si tiene conto del fatto che lo schema è piuttosto semplice, e che ci troviamo in un caso più che particolare; non sempre è possibile, infatti, arrivare ad un tale livello di ridondanza, che risulta essere tra i più bassi in assoluto (vi sono anche la quarta e la quinta forma normale, ma oltre al fatto che non siano oggetto di studio da parte del corso, raramente vengono applicate, poiché comportano un'eccessiva frammentazione dei dati con conseguente aumento della potenza computazionale richiesta per generare un'informazione di senso compiuto).

Implementazione:

Data Definition Language

Di seguito vengono riportate le istruzioni necessarie per la creazione delle tabelle nel database

```
CREATE TABLE DIPENDENTE(  
  NOME_DIP VARCHAR2(20) NOT NULL,  
  COGNOME_DIP VARCHAR2(20) NOT NULL,  
  CF_DIP CHAR(16) PRIMARY KEY,  
  DATA_NASCITA_DIP DATE NOT NULL,  
  NUMERO_TEL_DIP CHAR(10) NOT NULL,  
  CAP_DIP VARCHAR2(20),  
  CITTA_DIP VARCHAR2(20),  
  VIA_DIP VARCHAR2(20));
```

```
CREATE TABLE RICHIAMO(  
  CF_RIC CHAR(16) NOT NULL,  
  MOTIVAZIONE VARCHAR2(20) NOT NULL,  
  DATA_RIC DATE NOT NULL,  
  ENTITA NUMBER NOT NULL,  
  CONSTRAINT PK_RICHIAMO PRIMARY KEY (CF_RIC,DATA_RIC),  
  CONSTRAINT FK_RICHIAMO FOREIGN KEY (CF_RIC) REFERENCES DIPENDENTE(CF_DIP)  
  ON DELETE CASCADE,  
  CONSTRAINT Motive CHECK(MOTIVAZIONE IN('LOW_HOURS','RITARDO','CATTIVO')),  
  CONSTRAINT Entity CHECK (ENTITA BETWEEN 1 and 4)  
);
```

```
CREATE TABLE STIPENDIO(  
  CF_STIP CHAR(16) NOT NULL,  
  IMPORTO NUMBER NOT NULL,  
  TRATTENUTE NUMBER,  
  DATA_STIP DATE NOT NULL,  
  CONSTRAINT PK_STIP PRIMARY KEY (CF_STIP,DATA_STIP),  
  CONSTRAINT FK_STIP FOREIGN KEY (CF_STIP) REFERENCES DIPENDENTE(CF_DIP)  
  ON DELETE CASCADE);
```

```
CREATE TABLE PRESENZE(  
  CF_PRES CHAR(16) NOT NULL,  
  DATA_INIZIO_PRES DATE NOT NULL,  
  DATA_FINE_PRES DATE NOT NULL,  
  CONSTRAINT PK_PRES PRIMARY KEY (CF_PRES,DATA_INIZIO_PRES),  
  CONSTRAINT FK_PRES FOREIGN KEY (CF_PRES) REFERENCES DIPENDENTE(CF_DIP)  
  ON DELETE CASCADE  
);
```

```
CREATE TABLE NEGOZIO(  
  CAP_NEG VARCHAR2(20),  
  CITTA_NEG VARCHAR2(20),
```

```
VIA_NEG VARCHAR2(20),
QUARTIERE_NEG VARCHAR2(20) PRIMARY KEY,
DATA_INIZIO_NEG DATE NOT NULL,
CF_DIRIGENTE_NEG CHAR(16) NOT NULL,
CONSTRAINT FK_DIRIGENTE FOREIGN KEY (CF_DIRIGENTE_NEG) REFERENCES DIPENDENTE(CF_DIP)
);
```

```
CREATE TABLE LAVORA_IN(
DATA_IN_DIP DATE NOT NULL,
CF_LAVIN CHAR(16) NOT NULL,
QUARTIERE_LAVIN VARCHAR(20) NOT NULL,
CONSTRAINT FK_CFDIP FOREIGN KEY (CF_LAVIN) REFERENCES DIPENDENTE(CF_DIP)
ON DELETE CASCADE,
CONSTRAINT FK_QUART FOREIGN KEY (QUARTIERE_LAVIN) REFERENCES NEGOZIO(QUARTIERE_NEG),
CONSTRAINT PK_LAVORA_IN PRIMARY KEY(CF_LAVIN,QUARTIERE_LAVIN)
);
```

```
CREATE TABLE CLIENTE(
NUMERO_TEL_CLIENTE CHAR(10) PRIMARY KEY,
CAP_CLIENTE VARCHAR2(20),
CITTA_CLIENTE VARCHAR2(20),
VIA_CLIENTE VARCHAR2(20),
NOME_CLIENTE VARCHAR2(20) NOT NULL,
COGNOME_CLIENTE VARCHAR2(20) NOT NULL
);
```

```
CREATE TABLE TESSERA(
ID_TESSERA NUMBER PRIMARY KEY,
DATA_EMISSIONE_TESSERA DATE NOT NULL,
DATA_SCADENZA_TESSERA DATE NOT NULL,
NUMERO_TEL_TESSERA CHAR(10) NOT NULL,
QUARTIERE_TESSERA VARCHAR2(20) NOT NULL,
CONSTRAINT FK_NTEL FOREIGN KEY (NUMERO_TEL_TESSERA) REFERENCES
CLIENTE(NUMERO_TEL_CLIENTE),
CONSTRAINT FK_QUARTIERE FOREIGN KEY (QUARTIERE_TESSERA) REFERENCES
NEGOZIO(QUARTIERE_NEG)
);
```

```
CREATE TABLE CORRIERE(
NOME_CORRIERE VARCHAR2(20) NOT NULL,
CAP_CORRIERE VARCHAR2(20),
CITTA_CORRIERE VARCHAR2(20),
VIA_CORRIERE VARCHAR2(20),
PIVA_CORRIERE CHAR(11) PRIMARY KEY
);
```

```
CREATE TABLE CARRELLO(
ID_CARRELLO NUMBER PRIMARY KEY CHECK (ID_CARRELLO > 0),
ID_TESSERA_CARRELLO NUMBER,
CONSTRAINT FK_IDTESSERA FOREIGN KEY (ID_TESSERA_CARRELLO) REFERENCES TESSERA(ID_TESSERA)
);
```

```
CREATE TABLE SPEDIZIONE(  
ID_CARRELLO_SPEDIZIONE NUMBER NOT NULL,  
DATA_SPEDIZIONE DATE NOT NULL,  
TRACKING_SPEDIZIONE VARCHAR(30) NOT NULL,  
CAP_SPEDIZIONE VARCHAR2(20),  
CITTA_SPEDIZIONE VARCHAR2(20),  
VIA_SPEDIZIONE VARCHAR2(20),  
COSTO_SPEDIZIONE NUMBER NOT NULL,  
PIVA_SPEDIZIONE CHAR(11) NOT NULL,  
CONSTRAINT FK_PIVA FOREIGN KEY (PIVA_SPEDIZIONE) REFERENCES CORRIERE(PIVA_CORRIERE),  
CONSTRAINT PK_SPEDIZIONE PRIMARY KEY(TRACKING_SPEDIZIONE,PIVA_SPEDIZIONE),  
CONSTRAINT FK_ID_CARR_SPE FOREIGN KEY (ID_CARRELLO_SPEDIZIONE) REFERENCES  
CARRELLO(ID_CARRELLO)  
);
```

```
CREATE TABLE CASA_PROD(  
PIVA_CASAPROD CHAR(11) PRIMARY KEY,  
CAP_CASAPROD VARCHAR2(20),  
CITTA_CASAPROD VARCHAR2(20),  
VIA_CASAPROD VARCHAR2(20),  
NOME_CASAPROD VARCHAR2(20) NOT NULL  
);
```

```
CREATE TABLE PRODOTTO(  
CODICE_PRODOTTO CHAR(13) PRIMARY KEY,  
NOME_PRODOTTO VARCHAR(20) NOT NULL,  
PREZZO_PRODOTTO NUMBER NOT NULL CHECK (PREZZO_PRODOTTO > 0),  
PUNTI_PRODOTTO NUMBER NOT NULL CHECK (PUNTI_PRODOTTO > 0),  
CATEGORIA_PRODOTTO NUMBER NOT NULL CHECK (CATEGORIA_PRODOTTO BETWEEN 0 and 4),  
CP_PIVA_PRODOTTO CHAR(11) NOT NULL,  
CONSTRAINT FK_CP FOREIGN KEY (CP_PIVA_PRODOTTO) REFERENCES CASA_PROD(PIVA_CASAPROD)  
);
```

```
CREATE TABLE INCLUDE(  
ID_INCLUDE NUMBER,  
CODICE_INCLUDE CHAR(13),  
SCONTO_INCLUDE NUMBER NOT NULL CHECK (SCONTO_INCLUDE BETWEEN 1 AND 99),  
QUANTITY_INCLUDE NUMBER NOT NULL CHECK (QUANTITY_INCLUDE > 0),  
CONSTRAINT FK_ID FOREIGN KEY (ID_INCLUDE) REFERENCES CARRELLO(ID_CARRELLO),  
CONSTRAINT FK_PRODOTTO FOREIGN KEY (CODICE_INCLUDE) REFERENCES  
PRODOTTO(CODICE_PRODOTTO),  
CONSTRAINT PK_INCLUDE PRIMARY KEY(ID_INCLUDE,CODICE_INCLUDE)  
);
```

```
CREATE TABLE FORNITORE(  
NOME_FORNITORE VARCHAR2(20) NOT NULL,  
PIVA_FORNITORE CHAR(11) PRIMARY KEY,
```

```
CAP_FORNITORE VARCHAR2(20),
CITTA_FORNITORE VARCHAR2(20),
VIA_FORNITORE VARCHAR2(20));
CREATE TABLE FORNITURA(
TRACKING_FORNITURA VARCHAR(30) UNIQUE,
DATA_FORNITURA DATE,
PIVA_FORNITURA CHAR(11),
QUARTIERE_FORNITURA VARCHAR2(20),
CONSTRAINT FK_FORN FOREIGN KEY (PIVA_FORNITURA) REFERENCES FORNITORE(PIVA_FORNITORE),
CONSTRAINT FK_QUARTFORN FOREIGN KEY (QUARTIERE_FORNITURA) REFERENCES
NEGOZIO(QUARTIERE_NEG),
CONSTRAINT PK_FORNITURA PRIMARY KEY(TRACKING_FORNITURA,DATA_FORNITURA)
);
```

```
CREATE TABLE CONTENUTO_FORN(
CODICE_CONTFORN CHAR(13) NOT NULL,
DATA_FORNITURA_CONTFORN DATE NOT NULL,
TRACKING_CONTFORN VARCHAR(30) NOT NULL,
COSTO_CONTFORN NUMBER NOT NULL CHECK (COSTO_CONTFORN > 0),
QUANTITY_CONTFORN NUMBER NOT NULL CHECK (QUANTITY_CONTFORN > 0),
CONSTRAINT PK_CONTFORN PRIMARY
KEY(TRACKING_CONTFORN,DATA_FORNITURA_CONTFORN,CODICE_CONTFORN),
CONSTRAINT FK_CONT_PROD FOREIGN KEY (CODICE_CONTFORN) REFERENCES
PRODOTTO(CODICE_PRODOTTO),
CONSTRAINT FK_CONT_FORN FOREIGN KEY (TRACKING_CONTFORN) REFERENCES
FORNITURA(TRACKING_FORNITURA)
);
```

Data Manipulation Language

Di seguito sono riportate le istruzioni per il popolamento nel database

```
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('WBLGSF83S17H670C', 'ERMINIO', 'OTTONE', TO_DATE('08/07/1999', 'DD/MM/YYYY'),
'5924516732');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('SREFLV99P13F839B', 'FULVIO', 'SERAIO', TO_DATE('13/09/1999', 'DD/MM/YYYY'), '3465994994');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('TWPDGQ52P10G955I', 'LUCA', 'PEZZOLLA', TO_DATE('11/07/1997', 'DD/MM/YYYY'),
'5519558793');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('YFAQKD66C20A931A', 'SIMONE', 'MICILLO', TO_DATE('08/05/1995', 'DD/MM/YYYY'),
'8998750991');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('NLNKMP57C70H783K', 'LUIGI', 'ONOFRO', TO_DATE('02/01/1994', 'DD/MM/YYYY'), '9504960226');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('CMXTCB35T08G976F', 'GIUSEPPE', 'SALVE', TO_DATE('22/09/1986', 'DD/MM/YYYY'),
'0687925513');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('TOEGZZ33E23B462K', 'RAFFAELE', 'PICCOLOMONTE', TO_DATE('21/03/1988', 'DD/MM/YYYY'),
'0876579717');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('WMRZBM72H10L686K', 'JULIE', 'ARRIVATA', TO_DATE('15/08/1984', 'DD/MM/YYYY'),
'1139159418');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('NVNPGH79E18A041C', 'FRANCESCA', 'DONNA', TO_DATE('12/04/1987', 'DD/MM/YYYY'),
'3043012128');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('SBVCVL47B20E126I', 'FRANCESCO', 'MINESTRA', TO_DATE('08/02/1979', 'DD/MM/YYYY'),
'1980766008');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('ZPMNTC42T24L727M', 'ALESSIO', 'FERALE', TO_DATE('19/09/1979', 'DD/MM/YYYY'),
'4945514615');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('RBYLZF56L71A805E', 'TONY', 'TEAMARA', TO_DATE('29/09/1987', 'DD/MM/YYYY'), '092222006');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('VTGBZX28E69D271Z', 'ANGELO', 'CARMELLE', TO_DATE('13/03/1984', 'DD/MM/YYYY'),
'0129923415');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('VNSTFO97B54A149S', 'ARDENTE', 'POLLETTI', TO_DATE('13/10/1992', 'DD/MM/YYYY'),
'5091361721');
INSERT INTO DIPENDENTE(CF_DIP, NOME_DIP, COGNOME_DIP, DATA_NASCITA_DIP, NUMERO_TEL_DIP)
VALUES ('TZNCGW92P23H928A', 'LAURA', 'MAMELI', TO_DATE('11/10/1976', 'DD/MM/YYYY'),
'1121455737');
```

```

INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES ('FOREROTTA',
TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'SREFLV99P13F839B');
INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES
('SOMMAVULCANICA', TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'TWPDGQ52P10G955I');
INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES ('CAVERZANO',
TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'YFAQKD66C20A931A');
INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES ('DATABASE',
TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'RBYLZF56L71A805E');
INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES ('ALGEBRA',
TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'NLNKMP57C70H783K');
INSERT INTO NEGOZIO(QUARTIERE_NEG, DATA_INIZIO_NEG, CF_DIRIGENTE_NEG) VALUES
('SANTANASTAFRICA', TO_DATE('07/07/2020', 'DD/MM/YYYY'), 'SBVCVL47B20E126I');

```

```

INSERT INTO RICHIAMO(CF_RIC, MOTIVAZIONE, DATA_RIC, ENTITA) VALUES ('RBYLZF56L71A805E',
'RITARDO', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 1);

```

```

INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('SREFLV99P13F839B',
'26000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('TWPDGQ52P10G955I',
'26000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('YFAQKD66C20A931A',
'26000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('VTGBZX28E69D271Z', '30',
TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('CMXTCB35T08G976F',
'3000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 2999);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('TOEGZZ33E23B462K',
'8000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('RBYLZF56L71A805E',
'5500', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('SBVCVL47B20E126I',
'2300', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);
INSERT INTO STIPENDIO(CF_STIP, IMPORTO, DATA_STIP, TRATTENUTE) VALUES ('WBLGSF83S17H670C',
'10000', TO_DATE('08/07/2022', 'DD/MM/YYYY'), 0);

```

```

INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('SREFLV99P13F839B',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 20:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('TWPDPGQ52P10G955I',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 20:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('YFAQKD66C20A931A',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 20:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('NLNKMP57C70H783K',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 20:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('CMXTCB35T08G976F',
TO_DATE('07/07/2022 10:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 14:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('TOEGZZ33E23B462K',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 17:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('WMRZBM72H10L686K',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 17:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('NVNPGH79E18A041C',
TO_DATE('07/07/2022 09:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 16:00:00',
'DD/MM/YYYY HH24:MI:SS'));
INSERT INTO PRESENZE(CF_PRES, DATA_INIZIO_PRES, DATA_FINE_PRES) VALUES ('SBVCVL47B20E126I',
TO_DATE('07/07/2022 13:00:00', 'DD/MM/YYYY HH24:MI:SS'), TO_DATE('07/07/2022 20:00:00',
'DD/MM/YYYY HH24:MI:SS'));

```

```

INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'SREFLV99P13F839B', 'FOREROTTA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'TWPDPGQ52P10G955I', 'SOMMAVULCANICA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'YFAQKD66C20A931A', 'CAVERZANO');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'RBYLZF56L71A805E', 'DATABASE');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'NLNKMP57C70H783K', 'ALGEBRA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022',
'DD/MM/YYYY'), 'CMXTCB35T08G976F', 'FOREROTTA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022',
'DD/MM/YYYY'), 'TOEGZZ33E23B462K', 'FOREROTTA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022',
'DD/MM/YYYY'), 'WMRZBM72H10L686K', 'SOMMAVULCANICA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022',
'DD/MM/YYYY'), 'NVNPGH79E18A041C', 'SOMMAVULCANICA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('07/07/2020',
'DD/MM/YYYY'), 'SBVCVL47B20E126I', 'SANTANASTAFRICA');

```



```
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022', 'DD/MM/YYYY'), 'ZPMNTC42T24L727M', 'SANTANASTAFRICA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022', 'DD/MM/YYYY'), 'VTGBZX28E69D271Z', 'DATABASE');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022', 'DD/MM/YYYY'), 'VNSTFO97B54A149S', 'ALGEBRA');
INSERT INTO LAVORA_IN(DATA_IN_DIP, CF_LAVIN, QUARTIERE_LAVIN) VALUES (TO_DATE('30/01/2022', 'DD/MM/YYYY'), 'TZNCGW92P23H928A', 'ALGEBRA');
```

```
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889108', 'ALEXANDRA', 'ROTANTE');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889103', 'MARIAVIOLA', 'RICCIARDI');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889115', 'MARIO', 'BIANCHI');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889342', 'BRUCE', 'VOLZE');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889512', 'ROSANNA', 'AMATO');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889678', 'NELLO', 'CASTELLONE');
INSERT INTO CLIENTE(NUMERO_TEL_CLIENTE, NOME_CLIENTE, COGNOME_CLIENTE) VALUES ('1254889632', 'TONY', 'STRANO');
```

```
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (0, TO_DATE('08/03/2022', 'DD/MM/YYYY'), TO_DATE('08/03/2023', 'DD/MM/YYYY'), '1254889108', 'FOREROTTA');
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (1, TO_DATE('02/03/2022', 'DD/MM/YYYY'), TO_DATE('02/03/2023', 'DD/MM/YYYY'), '1254889342', 'SOMMAVULCANICA');
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (2, TO_DATE('05/03/2022', 'DD/MM/YYYY'), TO_DATE('05/03/2023', 'DD/MM/YYYY'), '1254889512', 'CAVERZANO');
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (3, TO_DATE('05/03/2022', 'DD/MM/YYYY'), TO_DATE('05/03/2023', 'DD/MM/YYYY'), '1254889678', 'ALGEBRA');
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (4, TO_DATE('05/03/2022', 'DD/MM/YYYY'), TO_DATE('05/03/2023', 'DD/MM/YYYY'), '1254889632', 'DATABASE');
INSERT INTO TESSERA(ID_TESSERA, DATA_EMISSIONE_TESSERA, DATA_SCADENZA_TESSERA, NUMERO_TEL_TESSERA, QUARTIERE_TESSERA) VALUES (5, TO_DATE('05/03/2022', 'DD/MM/YYYY'), TO_DATE('05/03/2023', 'DD/MM/YYYY'), '1254889115', 'SANTANASTAFRICA');
```

```
INSERT INTO CORRIERE(PIVA_CORRIERE, NOME_CORRIERE) VALUES ('68903970322', 'POSTE_ITALANE');
INSERT INTO CORRIERE(PIVA_CORRIERE, NOME_CORRIERE) VALUES ('60538180144', 'GLF');
INSERT INTO CORRIERE(PIVA_CORRIERE, NOME_CORRIERE) VALUES ('67904120789', 'BARTOLONE');
INSERT INTO CORRIERE(PIVA_CORRIERE, NOME_CORRIERE) VALUES ('69848540111', 'DHJ');
```

```
INSERT INTO CASA_PROD(PIVA_CASAPROD, NOME_CASAPROD) VALUES ('68903970959',
'UniversaMusicIta');
INSERT INTO CASA_PROD(PIVA_CASAPROD, NOME_CASAPROD) VALUES ('60538180427', 'SoniMusic');
INSERT INTO CASA_PROD(PIVA_CASAPROD, NOME_CASAPROD) VALUES ('67904120994', 'WarneMusicIta');
INSERT INTO CASA_PROD(PIVA_CASAPROD, NOME_CASAPROD) VALUES ('51714070870', 'Belive');
INSERT INTO CASA_PROD(PIVA_CASAPROD, NOME_CASAPROD) VALUES ('69848540154', 'TheOrchad');
```

```
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000001-Z', 'Bollicine',
50, 50, '68903970959', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000002-Z',
'LaVoceDellAdrone', 50, 50, '68903970959', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000003-Z',
'UnaDonnaPerNemico', 50, 50, '68903970959', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000004-Z',
'Lorenzo1995', 50, 50, '60538180427', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000005-Z',
'OvunqueProtegge', 50, 50, '60538180427', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000006-Z', 'Siberia', 50,
50, '60538180427', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000007-Z', 'LineaGoth',
50, 50, '67904120994', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000008-Z',
'BuonCompleannoElvis', 50, 50, '67904120994', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000009-Z',
'IBraviEICattivi', 50, 50, '67904120994', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000010-Z',
'MioFratelloUnico', 50, 50, '51714070870', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000011-Z',
'MicrochipEmozionante', 50, 50, '51714070870', 0);
```

```

INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000012-Z',
'NeroPerMeta', 50, 50, '51714070870', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000013-Z', 'Stanza', 50,
50, '69848540154', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000014-Z', 'Wow', 50,
50, '69848540154', 0);
INSERT INTO PRODOTTO(CODICE_PRODOTTO, NOME_PRODOTTO, PREZZO_PRODOTTO,
PUNTI_PRODOTTO, CP_PIVA_PRODOTTO, CATEGORIA_PRODOTTO) VALUES ('A-000000015-Z', 'Rimmell',
50, 50, '69848540154', 0);

```

```

INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('B2BGraffeti', '58574690861', '80100', 'FOREROTTA');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('Motiko', '37531470260', '80100', 'SOMMAVULCANICA');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('DeChecco', '26272180063', '80121', 'CASERTE');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('FashionLuxu', '22876910591', '80123', 'CAVERZANO');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('Favile', '15597970662', '80124', 'TELODO');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('NeapolitanBoutique', '35349810537', '80136', 'MUSEI');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('Nuagwe', '24067500884', '80112', 'VANVITELLO');
INSERT INTO FORNITORE(NOME_FORNITORE, PIVA_FORNITORE, CAP_FORNITORE, CITTA_FORNITORE)
VALUES ('DesjaVu', '24142410943', '80127', 'DENTE');

```

```

INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('adIdPfC0', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '58574690861',
'CAVERZANO');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('RjnnGsjv', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '37531470260',
'FOREROTTA');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('iMXCf7p', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '26272180063',
'SOMMAVULCANICA');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('y3a1Q8op', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '22876910591',
'CAVERZANO');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('QzhBJZWW', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '15597970662',
'FOREROTTA');

```

```
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('zKjF8BXv', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '35349810537',
'SOMMAVULCANICA');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('Usjqsoqz', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '24067500884',
'ALGEBRA');
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES ('TBVdEApv', TO_DATE('01/07/2022', 'DD/MM/YYYY'), '24142410943',
'DATABASE');
```

```
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000001-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'adIdPFc0', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000002-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'RjnnGsJv', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000003-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'iMXCFr7p', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000004-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'y3a1Q8op', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000005-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'QzhBJZWW', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000006-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'zKjF8BXv', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000007-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'Usjqsoqz', '1000', '30');
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES ('A-000000008-Z',
TO_DATE('01/07/2022', 'DD/MM/YYYY'), 'TBVdEApv', '1000', '30');
```

```
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (1,1);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (2,NULL);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (3,NULL);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (4,3);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (5,1);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (6,2);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (7,5);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (8,NULL);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (9,4);
INSERT INTO CARRELLO(ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES (10,NULL);
```

```

INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (1,
'A-000000001-Z', 0, 3);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (2,
'A-000000001-Z', 0, 6);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (3,
'A-000000002-Z', 0, 16);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (4,
'A-000000002-Z', 33, 4);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (5,
'A-000000003-Z', 0, 25);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (6,
'A-000000004-Z', 20, 7);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (7,
'A-000000005-Z', 15, 56);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (8,
'A-000000006-Z', 0, 100);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES (9,
'A-000000007-Z', 10, 33);
INSERT INTO INCLUDE(ID_INCLUDE, CODICE_INCLUDE, SCONTO_INCLUDE, QUANTITY_INCLUDE) VALUES
(10, 'A-000000008-Z', 5, 12);

```

```

INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (1,
TO_DATE('11/07/2022', 'DD/MM/YYYY'), 'KW2k1NiT', '80100', 'CAVERZANO', 'VIA FASULLA', 5,
'60538180144');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (2,
TO_DATE('09/07/2022', 'DD/MM/YYYY'), 'lQeAKIz5', '80115', 'FOREROTTA', 'VIA MOSTRA', 5,
'60538180144');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (2,
TO_DATE('05/07/2022', 'DD/MM/YYYY'), 'MLo20uqi', '80687', 'SOMMAVULCANICA', 'VIA ESPLOSIVA', 5,
'60538180144');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (3,
TO_DATE('11/07/2022', 'DD/MM/YYYY'), 'AU3cOnSn', '80118', 'CAVERZANO', 'VIA UMBERTO', 5,
'60538180144');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (4,
TO_DATE('02/05/2022', 'DD/MM/YYYY'), '4Fj8jgI5', '80142', 'FOREROTTA', 'VIA FASULLE', 5, '69848540111');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (5,
TO_DATE('13/03/2022', 'DD/MM/YYYY'), 'W7d1O5Mk', '80165', 'SOMMAVULCANICA', 'VIA FASULLA', 5,
'69848540111');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (6,
TO_DATE('23/01/2022', 'DD/MM/YYYY'), 'R6XPMfyt', '80789', 'ALGEBRA', 'VIA TOLEDE', 5, '69848540111');

```

```

INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (7,
TO_DATE('01/03/2022', 'DD/MM/YYYY'), 'NWAhtE8', '80422', 'ALGEBRA', 'VIA CHIAIE', 5, '69848540111');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (8,
TO_DATE('05/04/2022', 'DD/MM/YYYY'), '8FWcnQL5', '80987', 'DATABASE', 'VIA SAN GREGORIO', 5,
'69848540111');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES (9,
TO_DATE('11/06/2022', 'DD/MM/YYYY'), 'kID4GQ6A', '80111', 'DATABASE', 'VIA CARACCIULO', 5,
'69848540111');
INSERT INTO SPEDIZIONE(ID_CARRELLO_SPEDIZIONE, DATA_SPEDIZIONE, TRACKING_SPEDIZIONE,
CAP_SPEDIZIONE, CITTA_SPEDIZIONE, VIA_SPEDIZIONE, COSTO_SPEDIZIONE, PIVA_SPEDIZIONE) VALUES
(10, TO_DATE('10/06/2022', 'DD/MM/YYYY'), 'Flt3QJox', '80222', 'SANTANASTAFRICA', 'SPACCANAPULE', 5,
'69848540111');

```

Trigger:

Di seguito sono riportati i trigger utilizzati nel database

```

CREATE OR REPLACE trigger Can_Sell
BEFORE insert on INCLUDE
FOR EACH ROW

DECLARE
QUANTITY_INC NUMBER;
QUANTITY_MAG NUMBER;
CANT_SELL EXCEPTION;

BEGIN

SELECT COUNT(*) INTO QUANTITY_MAG FROM CONTENUTO_FORN WHERE :new.CODICE_INCLUDE =
CODICE_CONTFORN;
SELECT COUNT(*) INTO QUANTITY_INC FROM INCLUDE WHERE :new.CODICE_INCLUDE = CODICE_INCLUDE;

IF (QUANTITY_INC + :new.QUANTITY_INCLUDE) > QUANTITY_MAG
THEN RAISE CANT_SELL;
END IF;

EXCEPTION
WHEN CANT_SELL THEN
RAISE_APPLICATION_ERROR(-20004,'CANT SELL MORE THAN YOU HAVE');

END;

```

```

CREATE OR REPLACE TRIGGER Max_dip_tot
BEFORE INSERT ON DIPENDENTI
FOR EACH ROW

DECLARE
N_neg NUMBER;
N_dep NUMBER;
N_much EXCEPTION;

BEGIN

SELECT COUNT(*) INTO N_neg FROM NEGOZIO;
SELECT COUNT(*) INTO N_dep FROM DIPENDENTI;

IF (N_dep + 1) > (15*N_neg) THEN
RAISE N_much;
END IF;

EXCEPTION
WHEN N_much THEN
RAISE_APPLICATION_ERROR(-20009, 'TOO MANY WORKERS IN THE DB');

END;

```

```

CREATE OR REPLACE TRIGGER Dirigente
BEFORE INSERT OR UPDATE ON NEGOZIO
FOR EACH ROW

DECLARE
Data_Lav DATE;
Cant_manage EXCEPTION;

BEGIN
SELECT DATA_IN_DIP INTO Data_Lav FROM LAVORA_IN WHERE :new.CF_DIRIGENTE = CF_LAVIN;

IF (:new.DATA_INIZIO_NEG - Data_Lav) < 365
THEN RAISE Cant_manage;
END IF;

EXCEPTION
WHEN Cant_manage THEN
RAISE_APPLICATION_ERROR(-20007, 'CANT MANAGE, HE WORKED FOR LESS THAN A YEAR');

END;

```

```
CREATE OR REPLACE TRIGGER Pres_Max_Time  
BEFORE INSERT OR UPDATE ON PRESENZE  
FOR EACH ROW
```

```
DECLARE  
Presence NUMBER;  
Nine_hour EXCEPTION;  
Twelve_hour EXCEPTION;
```

```
BEGIN  
SELECT 24*(:new.DATA_FINE_PRES - :new.DATA_INIZIO_PRES) INTO Presence FROM dual;  
IF Presence >= 9 AND Presence < 12  
THEN RAISE Nine_hour;  
END IF;
```

```
IF Presence >= 12  
THEN RAISE Twelve_hour;  
END IF;
```

```
EXCEPTION  
WHEN Nine_hour THEN  
dbms_output.put_line('YOU ARE WORKING TOO MUCH! MAX HOURS FOR A DAY:8');
```

```
WHEN Twelve_hour THEN  
RAISE_APPLICATION_ERROR(-20008,'THIS TURN DOESNT COUNT, TOO MANY HOURS');
```

```
END;
```

```
CREATE OR REPLACE trigger Max_Richiame  
BEFORE insert on RICHIAMO  
FOR EACH ROW
```

```
DECLARE  
Max_R EXCEPTION;  
Richiami NUMBER;
```

```
BEGIN  
SELECT COUNT(*) INTO Richiami FROM RICHIAMO WHERE :new.cf_ric = cf_ric;  
IF Richiami > 9 THEN RAISE Max_R;  
END IF;
```

```
EXCEPTION  
WHEN Max_R THEN  
RAISE_APPLICATION_ERROR(-20002,'TOO MANY WARNINGS! NEED TO FIRE HIM?');
```

```
END;
```



```
CREATE OR REPLACE TRIGGER No_Double_Work
BEFORE INSERT ON LAVORA_IN
FOR EACH ROW
DECLARE
Double_Work EXCEPTION;
Count_Work NUMBER;
BEGIN
SELECT COUNT(*) INTO Count_Work FROM LAVORA_IN WHERE CF_LAVIN = :new.CF_LAVIN;
IF Count_Work > 0 THEN
RAISE Double_Work;
END IF;
EXCEPTION
WHEN Double_Work THEN
RAISE_APPLICATION_ERROR(-20001,'INVALID');
END;
```

```
CREATE OR REPLACE TRIGGER Stip_fix_dirig
BEFORE INSERT ON STIPENDIO
FOR EACH ROW

DECLARE
Low_money EXCEPTION;
CheckD NUMBER;

BEGIN
SELECT COUNT(*) INTO CheckD FROM NEGOZIO WHERE :new.CF_STIP = CF_DIRIGENTE_NEG;

IF CheckD > 0 THEN
IF :new.IMPORTO < 3200
THEN RAISE Low_money;
END IF;
END IF;

EXCEPTION
WHEN Low_money THEN
RAISE_APPLICATION_ERROR(-20006,'TOO LOW MONEY FOR HIM');
END;
```

```
CREATE OR REPLACE TRIGGER Stip_fix_dip
BEFORE INSERT ON STIPENDIO
FOR EACH ROW

DECLARE
Too_money EXCEPTION;
CheckD NUMBER;

BEGIN
SELECT COUNT(*) INTO CheckD FROM NEGOZIO WHERE :new.CF_STIP = CF_DIRIGENTE_NEG;

IF CheckD = 0 THEN
IF :new.IMPORTO > 1900
THEN RAISE Too_money;
END IF;
END IF;

EXCEPTION
WHEN Too_money THEN
RAISE_APPLICATION_ERROR(-20005,'TOO MUCH MONEY FOR HIM');
END;
```

```
CREATE OR REPLACE TRIGGER Max_Tessere
BEFORE INSERT ON TESSERA
FOR EACH ROW

DECLARE
Troppe_Tessere EXCEPTION;
Last_tessera DATE;

BEGIN
SELECT MAX(DATA_EMISSIONE_TESSERA) INTO Last_tessera FROM TESSERA WHERE
:new.NUMERO_TEL_TESSERA = NUMERO_TEL_TESSERA;

IF :new.DATA_EMISSIONE_TESSERA - Last_tessera < 365 THEN RAISE Troppe_Tessere;
END IF;

EXCEPTION
WHEN Troppe_Tessere THEN RAISE_APPLICATION_ERROR(-20003,'TOO MANY CARDS!');

END;
```

```

CREATE OR REPLACE TRIGGER Checketa
BEFORE INSERT ON DIPENDENTE
FOR EACH ROW

DECLARE
Eta_toomuch EXCEPTION;
Eta_toolow EXCEPTION;

BEGIN

IF (FLOOR((to_date(SYSDATE) - :new.DATA_NASCITA_DIP))/365) > 70 THEN
RAISE Eta_toomuch;
END IF;

IF (FLOOR((to_date(SYSDATE) - :new.DATA_NASCITA_DIP))/365) < 18 THEN
RAISE Eta_toolow;
END IF;

EXCEPTION

WHEN Eta_toomuch THEN
RAISE_APPLICATION_ERROR(-20012,'TOO OLD FOR THIS BUSINESS');

WHEN Eta_toolow THEN
RAISE_APPLICATION_ERROR(-20012,'TOO YOUNG FOR THIS BUSINESS');

END;

```

Procedure:

Di seguito sono riportate le procedure utilizzate nel database

```

CREATE OR REPLACE PROCEDURE Acquista (Tex NUMBER, Codprod CHAR, Qty NUMBER, Cartid NUMBER,
Discount NUMBER)
IS
Cart_max NUMBER;
Nomeprod VARCHAR2(30);
Wrong_tex EXCEPTION;
Wrong_Q EXCEPTION;
Wrong_Cart EXCEPTION;
Wrong_D EXCEPTION;
No_Prod EXCEPTION;
Flag NUMBER;
BEGIN
SELECT COUNT(*) INTO Flag FROM PRODOTTO WHERE CODICE_PRODOTTO = Codprod;

IF Flag < 1 THEN
RAISE No_Prod;
END IF;

```

```

IF Tex < 0 THEN
RAISE Wrong_tex;
END IF;

IF Qty < 0 THEN
RAISE Wrong_Q;
END IF;

IF Cartid < 0 THEN
RAISE Wrong_cart;
END IF;

IF Discount < 0 OR Discount > 99 THEN
RAISE Wrong_D;
END IF;

SELECT ID_CARRELLO INTO Cart_max FROM CARRELLO WHERE ID_CARRELLO = (SELECT MAX(ID_CARRELLO)
FROM CARRELLO);
Cart_Max := Cart_max +1;
IF Cartid = 0 THEN
    IF Tex = 0 THEN
        INSERT INTO CARRELLO (ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES(Cart_max,NULL);
        COMMIT;
    ELSE
        INSERT INTO CARRELLO (ID_CARRELLO, ID_TESSERA_CARRELLO) VALUES(Cart_max,Tex);
        COMMIT;
    END IF;
    dbms_output.put_line('CARRELLO NUMERO ' || TO_CHAR(Cart_Max) || ' CREATO');
END IF;

IF Cartid = 0 THEN
INSERT INTO INCLUDE (ID_INCLUDE,CODICE_INCLUDE,SCONTO_INCLUDE,QUANTITY_INCLUDE)
VALUES(Cart_max,Codprod,Discount,Qty);
COMMIT;
ELSE
INSERT INTO INCLUDE (ID_INCLUDE,CODICE_INCLUDE,SCONTO_INCLUDE,QUANTITY_INCLUDE)
VALUES(Cartid,Codprod,Discount,Qty);
COMMIT;
END IF;

SELECT NOME_PRODOTTO INTO Nomeprod FROM INCLUDE JOIN PRODOTTO ON CODICE_INCLUDE =
CODICE_PRODOTTO WHERE CODICE_PRODOTTO = Codprod GROUP BY NOME_PRODOTTO;
dbms_output.put_line('PRODOTTO ' || TO_CHAR(Nomeprod) || ' INSERITO NEL CARRELLO!');

EXCEPTION
WHEN No_Prod THEN
RAISE_APPLICATION_ERROR(-20029, 'THIS PRODUCT DOESNT EXIST IN THE DB');

WHEN Wrong_tex THEN
RAISE_APPLICATION_ERROR(-20030, 'CARD ID MUST BE GR8R THAN 0');

```

```

WHEN Wrong_Q THEN
RAISE_APPLICATION_ERROR(-20031, 'QUANTITY MUST BE GR8R THAN 0');

WHEN Wrong_CART THEN
RAISE_APPLICATION_ERROR(-20032, 'CART ID MUST BE GR8R THAN 0, OR AT LEAST 0');

WHEN Wrong_d THEN
RAISE_APPLICATION_ERROR(-20033, 'DISCOUNT MUST BE GR8R THAN 0 OR LESS THAN 100');

END Acquista;

```

```

CREATE OR REPLACE PROCEDURE Dip_Modello(Quart_Neg VARCHAR2)
IS
Codf CHAR(16);
Flag NUMBER;
Wrong_Q EXCEPTION;

BEGIN

SELECT COUNT(*) INTO Flag FROM NEGOZIO WHERE QUARTIERE_NEG = Quart_Neg;

IF Flag < 1 THEN
RAISE Wrong_Q;
END IF;
FOR x IN(select nome_dip as nome, cognome_dip as cognome, cf_dip as cf, ore
from
(select *
from dipendente join lavora_in on cf_dip = cf_lavin
where quartiere_lavin in (Quart_Neg)) join (select 24 * (data_fine_pres - data_inizio_pres) as ore, cf_pres
from presenze) on cf_lavin = cf_pres
where ore = (select max(ore) from(select *
from dipendente join lavora_in on cf_dip = cf_lavin
where quartiere_lavin in (Quart_Neg)) join (select 24 * (data_fine_pres - data_inizio_pres) as ore, cf_pres
from presenze) on cf_lavin = cf_pres) AND
cf_dip not in (select cf_ric from richiamo))
LOOP
Codf := x.cf;
INSERT INTO STIPENDIO(CF_STIP,IMPORTO,TRATTENUTE,DATA_STIP) VALUES
(Codf,1000,0,to_date(SYSDATE));
COMMIT;
END LOOP;

EXCEPTION
WHEN Wrong_Q THEN
RAISE_APPLICATION_ERROR(-20040, 'INVALID STORE');

END Dip_Modello;

```

```

CREATE OR REPLACE PROCEDURE Dip_Pessimo(Quart_Neg VARCHAR2)
IS
Codf CHAR(16);
Last_Stip DATE;
Flag NUMBER;
Wrong_Q EXCEPTION;

BEGIN

SELECT COUNT(*) INTO Flag FROM NEGOZIO WHERE QUARTIERE_NEG = Quart_Neg;

IF Flag < 1 THEN
RAISE Wrong_Q;
END IF;

FOR x IN(select nome_dip as nome, cognome_dip as cognome, cf_dip as cf, ore
from
(select *
from dipendente join lavora_in on cf_dip = cf_lavin
where quartiere_lavin in (Quart_Neg)) join (select 24 * (data_fine_pres - data_inizio_pres) as ore, cf_pres
from presenze) on cf_lavin = cf_pres
where ore = (select min(ore) from(select *
from dipendente join lavora_in on cf_dip = cf_lavin
where quartiere_lavin in (Quart_Neg)) join (select 24 * (data_fine_pres - data_inizio_pres) as ore, cf_pres
from presenze) on cf_lavin = cf_pres) AND
cf_dip in (select cf_ric from richiamo))
LOOP
Codf := x.cf;
SELECT MAX(DATA_STIP) INTO Last_Stip FROM STIPENDIO WHERE CF_STIP = x.cf;
INSERT INTO RICHIAMO (CF_RIC, MOTIVAZIONE, DATA_RIC, ENTITA) VALUES (x.cf,
'LOW_HOURS', to_date(SYSDATE), 1);
COMMIT;
UPDATE STIPENDIO

```

```
SET TRATTENUTE = TRATTENUTE + 300

WHERE DATA_STIP = Last_Stip AND CF_STIP = Codf;

COMMIT;

END LOOP;


EXCEPTION

WHEN Wrong_Q THEN

RAISE_APPLICATION_ERROR(-20040, 'INVALID STORE');


END Dip_Pessimo;
```

```
CREATE OR REPLACE PROCEDURE Rifornimento(Track VARCHAR2, Data DATE, Prodotto VARCHAR2,
Quantity NUMBER, Euros NUMBER, Quarter VARCHAR2, Piva VARCHAR2)
IS
prod_name VARCHAR2(30);
Flag NUMBER;
Flag_2 NUMBER;
Flag_3 NUMBER;
Flag_4 NUMBER;
Data_forn DATE;
Wrong_P EXCEPTION;
Wrong_Q EXCEPTION;
Wrong_E EXCEPTION;
Wrong_Quart EXCEPTION;
Wrong_Piva EXCEPTION;

BEGIN

SELECT COUNT(*) INTO Flag_2 FROM PRODOTTO WHERE CODICE_PRODOTTO = Prodotto;
IF Flag_2 < 1 THEN
RAISE Wrong_P;
END IF;

SELECT COUNT(*) INTO Flag_3 FROM NEGOZIO WHERE QUARTIERE_NEG = Quarter;
IF Flag_3 < 1 THEN
RAISE Wrong_Quart;
END IF;

SELECT COUNT(*) INTO Flag_4 FROM FORNITORE WHERE PIVA_FORNITORE = Piva;
IF Flag_4 < 1 THEN
RAISE Wrong_Piva;
END IF;
```

```

IF Quantity < 0 THEN
RAISE Wrong_Q;
END IF;

IF Euros < 0 THEN
RAISE Wrong_E;
END IF;

SELECT COUNT(*) INTO Flag FROM FORNITURA WHERE TRACKING_FORNITURA = Track;
IF Flag > 0 THEN
SELECT DATA_FORNITURA INTO Data_forn FROM FORNITURA WHERE TRACKING_FORNITURA = Track;
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES (Prodotto, Data_forn, Track,
Euros, Quantity);
COMMIT;

ELSE
INSERT INTO FORNITURA(TRACKING_FORNITURA, DATA_FORNITURA, PIVA_FORNITURA,
QUARTIERE_FORNITURA) VALUES (Track, Data, Piva, Quarter);
COMMIT;
INSERT INTO CONTENUTO_FORN(CODICE_CONFORN, DATA_FORNITURA_CONFORN,
TRACKING_CONFORN, COSTO_CONFORN, QUANTITY_CONFORN) VALUES (Prodotto, Data, Track, Euros,
Quantity);
COMMIT;
END IF;

SELECT NOME_PRODOTTO INTO prod_name FROM PRODOTTO WHERE CODICE_PRODOTTO = Prodotto;

IF Flag > 0 THEN
dbms_output.put_line('PRODOTTO ' || TO_CHAR(prod_name) || ' INSERITO NELLA FORNITURA '
|| TO_CHAR(Track) || '!');
ELSE
dbms_output.put_line('FORNITURA ' || TO_CHAR(Track) || ' CREATA! PRODOTTO
' || TO_CHAR(prod_name) || ' INSERITO!');
END IF;

EXCEPTION
WHEN Wrong_P THEN
RAISE_APPLICATION_ERROR(-20029, 'THIS PRODUCT DOESNT EXIST IN THE DB');

WHEN Wrong_Quart THEN
RAISE_APPLICATION_ERROR(-20040, 'INVALID STORE');

WHEN Wrong_Piva THEN
RAISE_APPLICATION_ERROR(-20050, 'INVALID MARIEUL');

WHEN Wrong_Q THEN
RAISE_APPLICATION_ERROR(-20031, 'QUANTITY MUST BE GR8R THAN 0');

WHEN Wrong_E THEN
RAISE_APPLICATION_ERROR(-20051, 'EUROS MUST BE GR8R THAN 0');
End Rifornimento;

```



```

CREATE OR REPLACE PROCEDURE Scontistica
IS
Codprod VARCHAR2(30);
BEGIN

FOR x IN(
SELECT CODICE_INCLUDE,QUANTITY_INCLUDE
FROM(SELECT *
      FROM TESSERA JOIN CARRELLO ON ID_TESSERA = ID_TESSERA_CARRELLO JOIN INCLUDE ON
      ID_CARRELLO = ID_INCLUDE)
GROUP BY CODICE_INCLUDE,QUANTITY_INCLUDE
HAVING QUANTITY_INCLUDE = (SELECT MIN(QUANTITY_INCLUDE) FROM(SELECT
CODICE_INCLUDE,QUANTITY_INCLUDE
FROM(SELECT *
      FROM TESSERA JOIN CARRELLO ON ID_TESSERA = ID_TESSERA_CARRELLO JOIN INCLUDE ON
      ID_CARRELLO = ID_INCLUDE)
GROUP BY CODICE_INCLUDE,QUANTITY_INCLUDE)))
LOOP
Codprod:= x.CODICE_INCLUDE;
UPDATE PRODOTTO
SET PREZZO_PRODOTTO = PREZZO_PRODOTTO * 0.7
WHERE CODICE_PRODOTTO = Codprod;
COMMIT;
END LOOP;

END Scontistica;

```

```

CREATE OR REPLACE PROCEDURE Sped_Tex (Cartid NUMBER, Datasped DATE, Cap VARCHAR2, City
VARCHAR2, Via VARCHAR2, Piva VARCHAR2, Track VARCHAR2)
IS
Cart_Price NUMBER;
Flag NUMBER;
Flag_2 NUMBER;
Wrong_Cart EXCEPTION;
Wrong_Piva EXCEPTION;

BEGIN
SELECT COUNT(*) INTO Flag FROM CARRELLO WHERE ID_CARRELLO = Cartid;

IF Flag < 1 THEN
RAISE Wrong_Cart;
END IF;

SELECT COUNT(*) INTO Flag_2 FROM CORRIERE WHERE PIVA_CORRIERE = Piva;
IF Flag_2 < 1 THEN
RAISE Wrong_Piva;

END IF;

```

```
SELECT 0 INTO Cart_Price FROM dual;
```

```
FOR x IN(  
SELECT NOME_PRODOTTO, MAX(PREZZO_PRODOTTO) AS PREZZO_PRODOTTO, MAX(QUANTITY_INCLUDE)  
AS QUANTITY_INCLUDE, MAX(SCONTO_INCLUDE) AS SCONTO_INCLUDE  
FROM (SELECT *  
FROM  
(SELECT *  
FROM CARRELLO JOIN TESSERA ON ID_TESSERA_CARRELLO = ID_TESSERA) JOIN INCLUDE ON ID_INCLUDE =  
ID_CARRELLO) JOIN PRODOTTO ON CODICE_INCLUDE = CODICE_PRODOTTO  
WHERE ID_CARRELLO = Cartid  
GROUP BY NOME_PRODOTTO)  
LOOP  
IF x.SCONTO_INCLUDE <> 0 THEN  
Cart_Price:= Cart_Price + ((x.PREZZO_PRODOTTO * x.SCONTO_INCLUDE) * x.QUANTITY_INCLUDE);  
ELSE  
Cart_Price:= Cart_Price + (x.PREZZO_PRODOTTO * x.QUANTITY_INCLUDE);  
END IF;  
END LOOP;
```

```
IF Cart_Price < 100 THEN  
INSERT INTO SPEDIZIONE  
(ID_CARRELLO_SPEDIZIONE,DATA_SPEDIZIONE,TRACKING_SPEDIZIONE,CAP_SPEDIZIONE,CITTA_SPEDIZION  
E,VIA_SPEDIZIONE,COSTO_SPEDIZIONE,PIVA_SPEDIZIONE)  
VALUES(Cartid,Datasped,Track,Cap,City,Via,5,Piva);  
COMMIT;  
ELSE  
INSERT INTO SPEDIZIONE  
(ID_CARRELLO_SPEDIZIONE,DATA_SPEDIZIONE,TRACKING_SPEDIZIONE,CAP_SPEDIZIONE,CITTA_SPEDIZION  
E,VIA_SPEDIZIONE,COSTO_SPEDIZIONE,PIVA_SPEDIZIONE)  
VALUES(Cartid,Datasped,Track,Cap,City,Via,0,Piva);  
COMMIT;  
END IF;
```

```
EXCEPTION  
WHEN Wrong_Cart THEN  
RAISE_APPLICATION_ERROR(-20050, 'THIS CART DOESNT EXIST');  
  
WHEN Wrong_Piva THEN  
RAISE_APPLICATION_ERROR(-20051, 'THIS COURRIER DOESNT EXIST');  
  
END Sped_Tex;
```