

PROGETTO BAR

Autore: D'ANGELO GIORGIO

Sommario

1 PROGETTAZIONE

- 1.1 Informazioni generali sul progetto
- 1.2 GLOSSARIO 1.3 ANALISI ENTITÀ

2 PROGETTAZIONE CONCETTUALE. DIAGRAMMA EE/R

- 2.1 DIAGRAMMA EE/R

3 PROGETTAZIONE LOGICA

- 3.1 DIAGRAMMA RELAZIONALE
- 3.2 Analisi tabelle

4 UTENTI ED OPERAZIONI

- 4.1 UTENTI 4.2 PRIVILEGI DI ACCESSO 4.3 TAVOLA DEGLI UTENTI 4.4 OPERAZIONI DEGLI UTENTI 4.5 VINCOLI DI INTEGRITÀ

5 NORMALIZZAZIONE

- 5.1 PRIMA FORMA NORMALE:
- 5.2 SECONDA FORMA NORMALE
- 5.3 TERZA FORMA NORMALE:

6 IMPLEMENTAZIONE

- 6.1 CREAZIONE UTENTI 6.2 DATA DEFINITION LANGUAGE 6.3 DATA MANIPULATION LANGUAGE 6.4 TRIGGER
- 6.5 FUNCTION
- 6.6 VISTE 6.7 DATA CONTROL LANGUAGE

1 PROGETTAZIONE

1.1 INFORMAZIONI GENERALI SUL PROGETTO

Si tratta della gestione di un bar dove vogliamo gestire le entrate e le uscite degli incassi.

In questo bar ci sono 3 gestori dove ad ognuno è assegnato un determinato compito, infatti c'è chi gestisce gli eventi, chi le uscite (carello) e chi le entrate(incasso).

Particolare attenzione è riservata sulla cassa in quanto non vogliamo che ci siano fuoriuscite di denaro non previste quindi ci sono delle regole ben severe per evitare che accada.

(In questo bar è registrato tutto il mese di gennaio).

1.2 Glossario

TERMINI	DESCRIZIONE	SINONIMI
GESTORE_BAR	Sono coloro che si occupano della gestione del bar	-
EVENTI	Eventi è l'entità in cui vengono espressi tutti gli eventi che può fare un bar con le varie promozioni	-
CARELLO	Si tratta delle spese che fa il bar per comprare i prodotti	-
FORNITORE	È colui che fornisce i prodotti al bar	-
DIPENDENTI	I vari dipendenti che lavorano per il bar	-
CONTRATTO	L'entità contratto sono i vari contratti che i gestori hanno fatto i dipendenti in cui viene precisato l'inizio e la scadenza del contratto	-
TURNO DI LAVORO	Viene gestito il turno dei dipendenti e gli incassi	-

1.3 Analisi entità e loro proprietà

Gestore Bar: Viene espresso la carta d'identità dei gestori e il loro ruolo in quanto in questo bar ogni gestore ha un proprio ruolo e si occupa quindi di una cosa specifica.

Eventi: Gli eventi vengono gestiti dai gestori con la rispettiva data e le varie promozioni.

Carello: Il carello ha due chiavi nome_prodotto e id_carello in questo modo in un singolo carello possono essere inseriti più prodotti e viene anche espressa la quantità di prodotti con i relativi costi.

Fornitura: La fornitura rifornisce il bar attraverso questi carelli la cui ha come chiave codice azienda per contraddistinguere i vari fornitori, c'è il nome del fornitore e che tipo di fornitore è se alimentare, di bevande ecc..

Dipendenti: Sono i dipendenti che lavorano nel bar in cui vengono contraddistinti da un codice fiscale come chiave e viene espresso il loro contratto di lavoro.

Contratto: Vengono gestiti i contratti dei dipendenti con la paga e questo dipendente che ruolo svolge nel bar se è un barista o cassiere e viene espressa la data di inizio e fine del suo contratto.

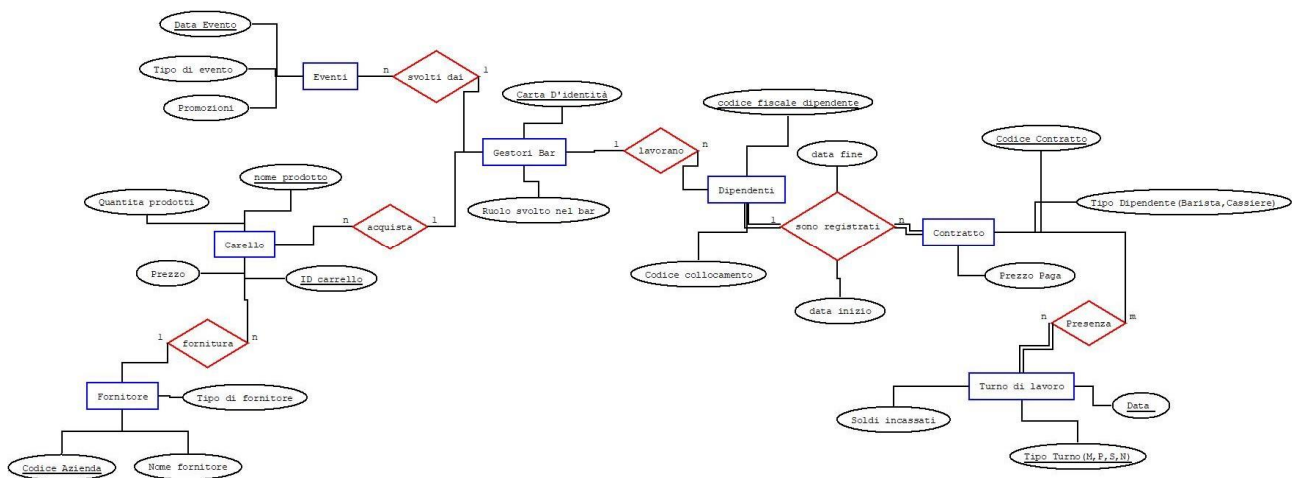
Turno di lavoro: Vengono gestiti i turni dei dipendenti con la data e la fascia oraria del giorno indicata con M=mattina P=pomeriggio S=sera N=notte e i soldi incassati.

2 Progettazione Concettuale. Diagramma E/R

2.1 Analisi entità e loro proprietà

Il Diagramma ER rappresenta una base di dati tramite associazioni, entità e attributi.

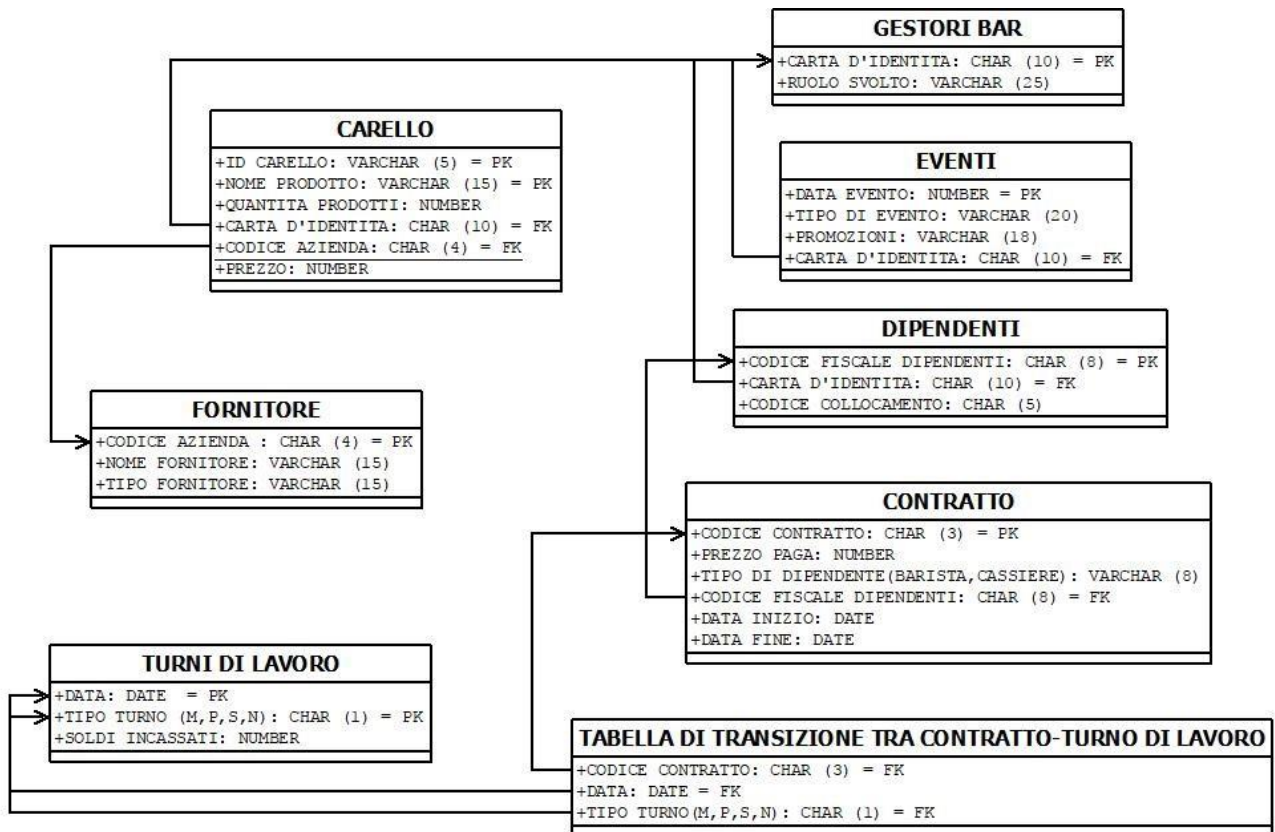
Il modello ER è il livello più astratto di una base di dati ed è colui che spiega astrattamente cosa accumuna i dati di una base di dati.



3 Progettazione Logica

Dopo il diagramma ER passiamo da uno schema concettuale ad uno di tipo logico e lo facciamo tramite lo schema relazione in cui traduciamo le varie entità come tabelle e gli attributi con le colonne e le varie associazioni con le chiavi esterne.

3.1 Diagramma Relazionale



3.2 Analisi delle tabelle

Adesso facciamo un'analisi di tutte le tabelle.

GESTORI_BAR= Questa tabella ha come chiave primaria CARTA_D_IDENTITA.

EVENTI =Questa tabella ha come chiave primaria DATA_EVENTO e ha come chiave esterna DATI_ANAGRAFICI che è riferito alla chiave primaria di Gestori_Bar ossia CARTA_D_IDENTITA.

CARELLO=Questa tabella ha una chiave composta cioè ha come chiavi ID_CARELLO e NOME_PRODOTTO e ha come chiavi esterne C_IDENTITA_GESTORE che è riferita alla chiave primaria di Gestori_bar ossia CARTA_D_IDENTITA e un'altra chiave esterna CODICE_AZIENDA1 che è riferita alla chiave primaria di FORNITORE ossia CODICE_AZIENDA.

FORNITORE=Questa tabella ha come chiave principale CODICE_AZIENDA.

DIPENDENTI=Questa tabella ha come chiave primaria CODICE_FISCALE_DIPENDENTI e ha come chiave esterna ID_CARTA_IDENTITA_PROPRIETARIO che è riferita alla chiave prima di Gestori_Bar ossia CARTA_D_IDENTITA.

CONTRATTO=Questa tabella ha come chiave principale CODICE_CONTRATTO e ha come chiave esterna ID_COD_DIPENDENTI che è riferita alla chiave primaria di DIPENDENTI ossia CODICE_FISCALE_DIPENDENTI

CONTRATTO-TURNO_DI_LAVORO=Si tratta di una tabella che nasce dall'associazione N:M tra contratto e turno_di_lavoro e ha come chiavi esterne le chiavi riferite a contratto e turno_di_lavoro ossia

ID_CODICE_CONTRATTO riferita alla chiave primare di contratto cioè CODICE_CONTRATTO e ha altre 2 chiavi esterne DATA1, TIPO_TURNO riferite alle chiavi primarie di turno_di_lavoro DATA, TIPO_DI_TURNO.

TURNO_DI_LAVORO=Si tratta di una tabella che contiene 2 chiavi quindi composta e sono DATA, TIPO_DI_TURNO.

4- Utenti ed Operazioni

Ora analizziamo la parte riguardante gli utenti del database, coloro che operano e i loro privilegi di accesso quindi trattiamo del DCL

4.1 UTENTI

Gli utenti che possono accedere al database sono 3 e ognuno avrà delle funzionalità diverse:

- ADMIN
- GESTORE BAR
- DIPENDENTI

4.2 Privilegi di accesso

Andiamo ad analizzare ciascuna categoria di utente ed i loro privilegi di accesso:

Admin

L'admin è colui che ha tutti i privilegi tra cui la creazione , la modifica e l'eliminazioni delle tabelle

Dipendenti

I dipendenti hanno la facoltà solo di vedere i turni di lavoro e i loro contratti

Gestore Bar

Ha tutti i privilegi tranne la cancellazione o creazioni di tabelle e non può modificare gli incassi se il giorno in questione è già passato per evitare eventuali truffe non solo dei dipendenti ma anche degli stessi in quanti essi sono 3.

4.3 Tavola degli utenti

Nella tavola degli utenti sono riportati i privilegi degli utenti precedentemente elencati.

UTENTE	VOLUME	PERMESSI
	3	

GESTORE BAR		create user gestorebar identified by admin; grant connect,create session to gestorebar; grant insert,update,delete on eventi to gestorebar; grant insert,update,delete on dipendenti to gestorebar; grant insert,update,delete on contratto to gestorebar; grant select,update on turno_di_lavoro to gestorebar; grant insert,update,delete on carello to gestorebar; grant select on fornitura to gestorebar;
ADMIN	1	ALL
DIPENDENTI	8	create user dipendenti identified by dipendenti1; grant connect,create session to dipendenti; grant select on accesso_limitato_dipendenti to dipendenti; grant select on dipendenti to dipendenti; grant select on contratto to dipendenti;

4.4 Operazioni degli utenti

Notiamo le operazioni che possono fare i due utenti all'interno di questo database.

Il gestore tramite delle function riesce a calcolare l'incasso del bar inserendo come valori di input nella function due date che indicano l'intervallo di tempo in cui sono calcolati gli incassi.

Esempio dal 1 gennaio fino al 30 gennaio vogliamo sapere quanto ho guadagnato tramite la function calcola_somma_incassi possiamo saperlo.

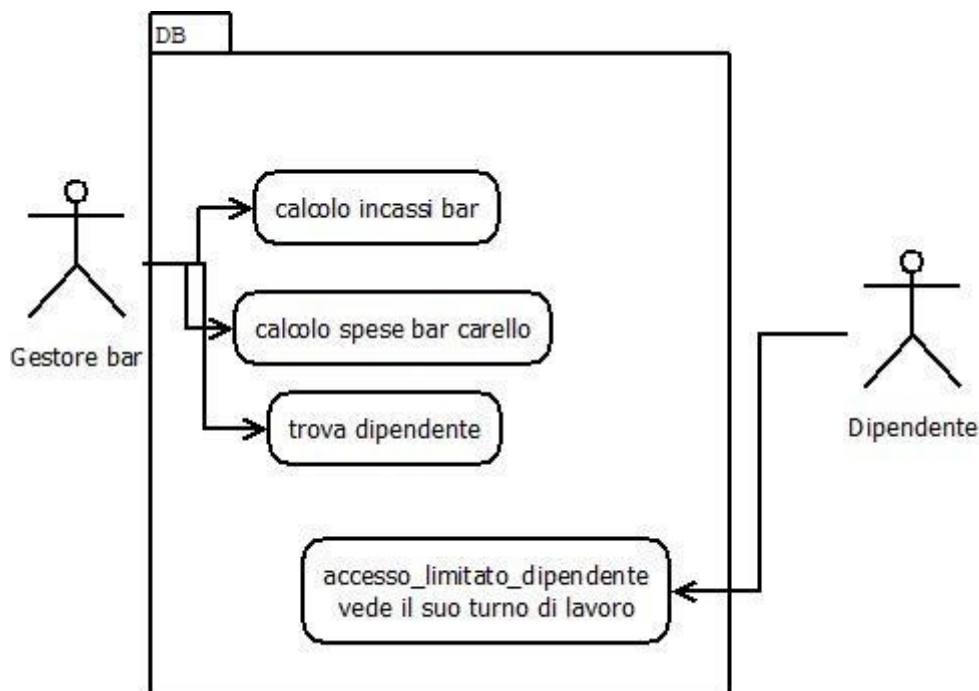
Il gestore può anche calcolare le spese del suo bar che sono dovute all'acquisto dei prodotti. Si tratta di una function che dato il numero del carello restituisce il costo complessivo del carello

Calcola_spese_carello.

Un'ultima funzione attribuita al gestore è quella di trovare il dipendente, questa è la funzione più importante si tratta di un function che dati in input due valori di cui uno è la data e l'altro è la fascia oraria siamo in grado di trovare il dipendente che lavora quel giorno.

E' molto importante soprattutto per quanto riguarda il dipendente cassiere infatti se nel caso in cui ci mancassero dei soldi nel bar sappiamo in quel determinato giorno e in quella fascia oraria chi stava lavorando con la function trova_dipendente.

Per quanto riguarda invece l'utente dipendente notiamo che l'unica opzione che può fare è quella di visualizzare la tabella turno_di_lavoro anche se in realtà è stata costruita una vista in modo che il dipendente non possa guardare gli incassi che fa il bar ma può guardare solo i suoi turni di lavoro.



4.5 Vincoli di integrità

Vincoli di integrità statici:

- Un dipendente può essere solo barista o cassiere.
- Il turno di lavoro può essere solo M(mattina) P(pomeriggio) S(sera) N(notte)..

Vincoli di integrità dinamici:

- Non può essere modificata o cancellata la data di un incasso se non è quello stesso giorno.
- Ci può essere un evento solo se passano almeno 7 giorni.
- Non si possono comprare più di 10 prodotti in uno stesso carello.

5 Normalizzazione

Nella verifica di normalità si combatte la ridondanza attraverso la caccia alle dipendenze funzionali anomale

5.1 Prima forma normale

La prima forma normale richiede che venga rispettato il vincolo di atomicità degli attributi. Lo schema rispetta questo requisito, in quanto non sono presenti né attributi composti né multivalore.

5.2 Seconda forma normale

Uno schema rispetta la seconda forma normale quando, nel caso siano presenti chiavi multiattributo, gli attributi abbiano una dipendenza funzionale completa dalla chiave. All'interno dello schema sono presenti più volte chiavi multi-attributo, ma in nessun caso i restanti attributi formano dipendenza parziale con gli attributi chiave, quindi possiamo affermare che la seconda forma normale è rispettata.

5.3 Terza forma normale

La terza forma normale richiede che gli attributi della relazione dipendano funzionalmente soltanto dalla chiave e non da attributi non chiave. Questo vincolo è anch'esso rispettato.

6 Implementazione

Terminata la fase logica si passa alla fase implementativa che riguarda quindi l'SQL tuttavia dobbiamo sottolineare che questo linguaggio dentro di sé ne racchiude molti altri e ognuno è utilizzato per dettagli implementativi specifici.

DDL permette la creazione, la modifica o la cancellazione delle tabelle

DML permette l'inserimento, l'aggiornamento o la cancellazione dei dati inseriti nelle tabelle

DCL permette di gestire i permessi degli utenti

PLSQL è un'estensione del SQL che ci permette di gestire funzioni, procedure e trigger in modo da avere un database più dinamico.

6.1 Creazione utenti

In questa fase creiamo gli utenti dopo essere loggati al database come amministratori.

```
-- Creazione Utenti
create user admin identified by admin1; create
user gestorebar identified by gestorebar1; create
user dipendenti identified by dipendenti1;
```


6.2 Data Definition Language

Lo schema del database è creato con il linguaggio DDL, che permette di creare, modificare ed eliminare tabelle e possiamo anche esprimere all'interno di essere dei vincoli statici.

--DROP TABLE

```
DROP TABLE GESTORI_BAR CASCADE CONSTRAINTS;
DROP TABLE EVENTI CASCADE CONSTRAINTS;
DROP TABLE DIPENDENTI CASCADE CONSTRAINTS;
DROP TABLE CONTRATTO CASCADE CONSTRAINTS;
DROP TABLE TURNO_DI_LAVORO CASCADE CONSTRAINTS;
DROP TABLE CONTRATTO_TURNO CASCADE CONSTRAINTS;
DROP TABLE FORNITURA CASCADE CONSTRAINTS;
DROP TABLE CARELLO CASCADE CONSTRAINTS;
```

TABELLE

--GESTORI_BAR

```
CREATE TABLE GESTORI_BAR
(CARTA_D_IDENTITA CHAR(10) PRIMARY KEY,
RUOLO_SVOLTO VARCHAR (25));
```

--EVENTI

```
CREATE TABLE EVENTI
(DATA_EVENTO DATE PRIMARY KEY,
TIPO_DI_EVENTO VARCHAR(20) NOT NULL,
PROMOZIONI VARCHAR(18),
DATI_ANAGRAFICI CHAR(10),
CONSTRAINT FK1 FOREIGN KEY(DATI_ANAGRAFICI)
REFERENCES GESTORI_BAR (CARTA_D_IDENTITA));
```

--DIPENDENTI

```
CREATE TABLE DIPENDENTI
(CODICE_FISCALE_DIPENDENTI CHAR(8) PRIMARY KEY,
CODICE_COLLOCAMENTO CHAR (5) NOT NULL,
ID_CARTA_IDENTITA_PROPRIETARIO CHAR(10),
CONSTRAINT FK2 FOREIGN KEY(ID_CARTA_IDENTITA_PROPRIETARIO)
REFERENCES GESTORI_BAR(CARTA_D_IDENTITA));
```

--CONTRATTO

```
CREATE TABLE CONTRATTO
(CODICE_CONTRATTO CHAR(3) PRIMARY KEY,
PREZZO_PAGA NUMBER,
TIPO_DI_DIPENDENTE VARCHAR(8),
DATA_INIZIO DATE,
DATA_FINE DATE,
ID_COD_DIPENDENTI CHAR (8),
CONSTRAINT V1 CHECK(TIPO_DI_DIPENDENTE IN ('BARISTA','CASSIERE')),
CONSTRAINT FK3 FOREIGN KEY(ID_COD_DIPENDENTI)
```

```
REFERENCES DIPENDENTI(CODICE_FISCALE_DIPENDENTI));
```

--TURNO_DI_LAVORO

```
CREATE TABLE TURNO_DI_LAVORO  
(DATA DATE,  
TIPO_DI_TURNO CHAR(1),  
SOLDI_INCASSATI NUMBER,  
CONSTRAINT V2 CHECK(TIPO_DI_TURNO IN('M','P','S','N')), CONSTRAINT  
PK PRIMARY KEY(DATA, TIPO_DI_TURNO));
```

--CONTRATTO_TURNO

```
CREATE TABLE CONTRATTO_TURNO (  
ID_CODICE_CONTRATTO CHAR(3),  
DATA1 DATE,  
TIPO_TURNO CHAR(1),  
CONSTRAINT FK6 FOREIGN KEY(ID_CODICE_CONTRATTO)  
REFERENCES CONTRATTO(CODICE_CONTRATTO),  
CONSTRAINT FK7 FOREIGN KEY(DATA1, TIPO_TURNO)  
REFERENCES TURNO_DI_LAVORO(DATA, TIPO_DI_TURNO));
```

--FORNITURA

```
CREATE TABLE FORNITURA  
(CODICE_AZIENDA CHAR(4) PRIMARY KEY,  
NOME_FORNITORE VARCHAR(15),  
TIPO_DI_FORNITORE VARCHAR(15));
```

--CARELLO

```
CREATE TABLE CARELLO  
(ID_CARELLO VARCHAR(5),  
NOME_PRODOTTO VARCHAR(25),  
QUANTITA_DI_PRODOTTI NUMBER,  
PREZZO NUMBER,  
C_IDENTITA_GESTORE CHAR(10),  
CODICE_AZIENDA1 CHAR (4),  
CONSTRAINT FK4 FOREIGN KEY(C_IDENTITA_GESTORE)  
REFERENCES GESTORI_BAR(CARTA_D_IDENTITA),  
CONSTRAINT FK5 FOREIGN KEY(CODICE_AZIENDA1)  
REFERENCES FORNITURA(CODICE_AZIENDA),  
CONSTRAINT PK1 PRIMARY KEY(ID_CARELLO, NOME_PRODOTTO));
```

6.3 Data Manipulation Language

Il DML è il linguaggio utilizzato per inserire modificare o cancellare i dati all'interno di un tabella.

```
-- POPOLAMENTO TOTALE
```

--GESTORI BAR

```
insert into gestori_bar(CARTA_D_IDENTITA,RUOLO_SVOLTO) values('AX759OXA22','GESTIONE EVENTI');
insert into gestori_bar(CARTA_D_IDENTITA,RUOLO_SVOLTO) values('BC932EQV4','GESTIONE CARELLO');
insert into gestori_bar(CARTA_D_IDENTITA,RUOLO_SVOLTO) values('CX759OXA22','GESTIONE
DIPENDENTI');
```

--EVENTI

```
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('13-GEN-
2017','MUSICA HOUSE','OFFERTA BIRRA','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('20-GIU-
2017','FREE BAR','SHOTS 1 EURO','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('27-
GIU2017','SUMMER','APERITIVI FREE','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('1-AGO-
2017','MAMACITA','CORNETTO+CAFFE 1 EURO','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('12-AGO-
2017','REVOLUTION','TENNIS 2 EURO','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('18-AGO-
2017','SERATA ROCK','FREE BAR','AX759OXA22');
INSERT INTO EVENTI(DATA_EVENTO,TIPO_DI_EVENTO,PROMOZIONI,DATI_ANAGRAFICI) VALUES('27-
AGO2017','APERITIVO','APEROL FREE','AX759OXA22');
```

--DIPENDENTI insert

into

```
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('DNGGRG96','12345','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('SERTCU90','12346','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('RETECV95','12347','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('NOIJHK88','12348','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('PORETV82','12349','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('REQASD75','12350','CX759OXA22'); insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('GERFDS99','12351','CX759OXA22');
insert into
dipendenti(CODICE_FISCALE_DIPENDENTI,CODICE_COLLOCAMENTO,ID_CARTA_IDENTITA_PROPRIETARIO)
values('DSAQWE75','12352','CX759OXA22');
```

-- CONTRATTO

insert into

```
contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI
PENDENTI) values('C01','200','CASSIERE','01-GEN-2016','01-GEN-2017','DNGGRG96'); insert into
contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI
PENDENTI) values('C02','200','CASSIERE','04-FEB-2016','04-MAR-2017','SERTCU90'); insert into
```

contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('C03','200','CASSIERE','07-GIU-2016','22-MAR-2017','RETECV95'); insert into contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('C04','300','CASSIERE','29-DIC-2016','30-GEN-2018','NOIJHK88'); insert into contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('B01','400','BARISTA','18-FEB-2017','22-SET-2019','PORETV82'); insert into contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('B02','450','BARISTA','06-AGO-2015','07-GEN-2017','REQASD75'); insert into contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('B03','250','BARISTA','02-SET-2014','25-DIC-2018','GERFDS99'); insert into contratto(CODICE_CONTRATTO,PREZZO_PAGA,TIPO_DI_DIPENDENTE,DATA_INIZIO,DATA_FINE,ID_COD_DI PENDENTI) values('B04','350','BARISTA','15-NOV-2016','11-APR-2019','DSAQWE75');

--TURNO DI LAVORO

insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('01-GEN-2017','M','120'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('01-GEN-2017','P','300'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('01-GEN-2017','S','500'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('01-GEN-2017','N','220'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('02-GEN-2017','M','150'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('02-GEN-2017','P','275'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('02-GEN-2017','S','325'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('02-GEN-2017','N','130'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('03-GEN-2017','M','100'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('03-GEN-2017','P','50'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('03-GEN-2017','S','199'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('03-GEN-2017','N','300'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('04-GEN-2017','M','626'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('04-GEN-2017','P','540'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('04-GEN-2017','S','147'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('04-GEN-2017','N','258'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('05-GEN-2017','M','362'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('05-GEN-2017','P','489'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('05-GEN-2017','S','920'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('05-GEN-2017','N','15'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('06-GEN-2017','M','69'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('06-GEN-2017','P','123'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('06-GEN-2017','S','250'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('06-GEN-2017','N','74'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('07-GEN-2017','M','140'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('07-GEN-2017','P','358'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('07-GEN-2017','S','210'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('07-GEN-2017','N','100'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('08-GEN-2017','M','75'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('08-GEN-2017','P','79'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('08-GEN-2017','S','178'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('08-GEN-2017','N','80'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('09-GEN-2017','M','397'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('09-GEN-2017','P','149'); insert into turno_di_lavoro(DATA,TIPO_DI_TURNO,SOLDI_INCASSATI) values('09-GEN-2017','S','39'); insert into

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

insert into contratto_turno(ID_CODICE_CONTRATTO,DATA1,TIPO_TURNO) values('B01','31-GEN-2017','M');
insert into contratto_turno(ID_CODICE_CONTRATTO,DATA1,TIPO_TURNO) values('B02','31-GEN-2017','P');
insert into contratto_turno(ID_CODICE_CONTRATTO,DATA1,TIPO_TURNO) values('B03','31-GEN-2017','S');
insert into contratto_turno(ID_CODICE_CONTRATTO,DATA1,TIPO_TURNO) values('B04','31-GEN-2017','N');

```

-- FORNITURA

```

insert into fornitura(CODICE_AZIENDA,NOME_FORNITORE,TIPO_DI_FORNITORE)
values('AA1','KIMBO','ALIMENTARE');
insert into fornitura(CODICE_AZIENDA,NOME_FORNITORE,TIPO_DI_FORNITORE) values('AA2','INGROSS
CASORIA','BEVANDE');
insert into fornitura(CODICE_AZIENDA,NOME_FORNITORE,TIPO_DI_FORNITORE) values('AA3','INGROSSO
BIBITE','ALIMENTARE');
insert into fornitura(CODICE_AZIENDA,NOME_FORNITORE,TIPO_DI_FORNITORE)
values('AA4','IANNACONE','ALIMENTARE');

```

-- CARELLO insert

```

into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C1','caffè kimbo','15','50','BC932EQV4','AA1'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C1','acqua lete','25','10','BC932EQV4','AA2'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C1','bevande alcoliche','10','150','BC932EQV4','AA3');
insert into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C2','cioccolata','20','50','BC932EQV4','AA1'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C2','pastecceria','30','21','BC932EQV4','AA4'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C2','rosticceria','70','35','BC932EQV4','AA4'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C3','cocacola','100','40','BC932EQV4','AA2'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C3','birre','60','30','BC932EQV4','AA2'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(

```

```

'C3','fanta','20','9','BC932EQV4','AA2'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C4','caffè kimbo','20','7','BC932EQV4','AA1'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C4','patatine','10','5','BC932EQV4','AA4'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C5','pasticceria','70','35','BC932EQV4','AA4'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C5','rosticceria','50','20','BC932EQV4','AA4'); insert
into
carello(ID_carello,nome_prodotto,quantita_di_prodotti,prezzo,C_identita_gestore,codice_azienza1)values(
'C6','birre','20','7','BC932EQV4','AA2');

COMMIT;

```

6.4 Trigger

I trigger DML sono utilizzati per implementare i vincoli dinamici i quali servono quando andiamo ad inserire cancellare o modificare dati all'interno di una tabella, se una certa condizione è violata allora succede una determinata cosa e sono essenziali per rendere un database dinamico.

Trigger 1:

--Il primo trigger è utilizzato per evitare truffe da eventuali dipendenti o dagli stessi gestori del bar se casomai si volessero truffare a vicenda e consiste in questo:

Non si possono né modificare né eliminare gli incassi se quest'ultimi è stato fatto un giorno precedente a quello corrente. Quindi ad esempio un incasso del giorno prima non si può modificare ecc..

```

create or replace trigger incassi_non_modificabili
BEFORE update or delete of soldi_incassati on turno_di_lavoro for each row declare
ultima_data date; errore exception; BEGIN select max(data) into ultima_data from
turno_di_lavoro; if CURRENT_DATE>ultima_data then raise errore; end if; exception
when errore then raise_application_error(-20001,'non puoi modificare o cancellare
gli incassi dei giorni precedenti');

```

```

end;

```

Trigger 2:

--Il secondo trigger riguarda una regola di business nel bar che dice che si può effettuare un altro evento se e solo se sono passati almeno 7 giorni dall'ultimo

```
create or replace trigger eventi_limitati
before insert or update of data_evento on
eventi for each row DECLARE errore
exception;

BEGIN

if current_date<(:old.data_evento + 7)
then raise errore;
end if;
exception
when errore
then
raise_application_error(-20002,'MASSIMO 1 EVENTO A SETTIMANA');

end;
```

Trigger 3:

--Un'altra regola di business all'interno del bar è che nn si possono comprare più di 10 prodotti all'interno dello stesso carrello

```
create or replace trigger carrello_pieno
BEFORE insert on carrello for each row
declare
max_prodotti    exception;
conta_prodotti  number;

begin
select count(nome_prodotto) into conta_prodotti from carrello
where id_carrello=:new.id_carrello;

if conta_prodotti>10 then raise
max_prodotti;
end if; exception
when max_prodotti then
raise_application_error(-20003,'Il carrello è pieno');    end;
```

6.5 Function

Attraverso l'utilizzo delle funzioni si cerca di automatizzare il più possibile il DB.

FUNCTION 1:

--Questa function dati in input due date restituisce in quell'arco di tempo l'incasso del bar (in questo database è registrato solo tutto il mese di gennaio quindi una data compresa da 01-gen-2017 fino al 31-gen-2017)

```
create or replace function calcolo_somma_incassi (data1 date,data2
date) return number as
```

```
totale
number; data
date; begin
select sum(soldi_incassati) into
totale from turno_di_lavoro
where data>data1 and
data<data2; return totale; end;
```

FUNCTION 2:

--Questa function trova_dipendente dato in input la data e la fascia oraria quindi M=mattina,P=pomeriggio,S=sera,N=notte trova quale dipendente stava lavorando quel giorno(la data deve essere un numero compreso tra il 1° e il 31 di gennaio e la fascia oraria appunto M o P o S o N.)

```
create or replace function trova_dipendente(datain date,fascia_oraria
varchar)
return varchar is
Cursor c1 is select id_cod_dipendenti from
(contratto join contratto_turno on
codice_contratto=id_codice_contratto)
join turno_di_lavoro on data=data1 and tipo_di_turno=tipo_turno where
data=datain and tipo_di_turno=fascia_oraria; nome varchar(10); Begin open c1;
LOOP
fetch c1 into nome; exit when
c1%NOTFOUND; END LOOP;
return nome; close c1;
```

end trova_dipendente;

FUNCTION 3:

--Questa function inserito il numero del carello ci permette di sapere il prezzo del carello totale.

```
create or replace function calcola_spese_carello(numero_carello varchar) return number
as soldi_spesi number; begin
select sum(prezzo) into soldi_spesi from carello
where id_carello=numero_carello;
return soldi_spesi; end;
```

6.6 Viste

Una vista può essere definita come l'alias di una query, ovvero serve per dare un nome ad un'interrogazione e trattare quest'ultima come una tabella tuttavia c'è una sostanziale differenza tra una tabella e una vista è che essa non viene salvata all'interno del database a meno che non si salvi all'interno del database come una vista materializzata.

Essenzialmente serve per gestire i privilegi di accesso di alcuni utenti in una tabella in cui non vogliamo rendere tutto visibile oppure per facilitare la scrittura delle query.

Vista 1:

--Questa vista è stata creata in quanto vogliamo limitare le informazioni che possono vedere i dipendenti tra cui l'incasso del bar in quanto sulla stessa entità ossia turno_di_lavoro non solo ci sono i turni di lavoro dei dipendenti ma c'è anche l'incasso quindi permettiamo attraverso questa vista di far sì che i dipendenti possano vedere le loro fasce orarie ma non gli permettiamo di visualizzare gli incassi del bar.

```
create materialized view accesso_limitato_dipendenti as
(select data,tipo_di_turno,id_cod_dipendenti
from (contratto join contratto_turno on id_codice_contratto=codice_contratto) join
turno_di_lavoro on
data=data1 and tipo_di_turno=tipo_turno);
```

6.7 Data Control Language

Il DCL viene utilizzato per assegnare i permessi ai vari utenti .

Permessi gestore bar

```
grant connect,create session to gestorebar; grant
insert,update,delete on eventi to gestorebar; grant
insert,update,delete on dipendenti to gestorebar; grant
```



```
insert,update,delete on contratto to gestorebar; grant  
select,update on turno_di_lavoro to gestorebar; grant  
insert,update,delete on carello to gestorebar; grant select on  
fornitura to gestorebar;
```

Permessi dipendenti

```
grant connect,create session to dipendenti;  
grant select on accesso_limitato_dipendenti to dipendenti; grant select on  
dipendenti to dipendenti;  
grant select on contratto to dipendenti;
```

Permessi Admin

```
grant all privileges to admin;
```

Quindi notiamo che l'admin ha tutti i privilegi, mentre i gestori del bar sono limitati in quando non possono né creare, né modificare ,né eliminare le tabelle e soprattutto non sono in grado di modificare gli incassi dei giorni precedenti a quello corrente in quanto anche gli stessi potrebbero rubare i soldi del bar. Infine ci sono i dipendenti i quali sono molti limitati,infatti hanno un accesso limitato sulla tabella turno_di_lavoro tramite una vista in cui loro possono vedere solo i loro orari di lavoro ma non l'incasso del bar e possono visualizzare i loro contratti sull'entità contratto.