

Gianpio Silvestri 0512107818

# Progetto Basi di Dati

## Catena di Negozi di Videogiochi

## Raccolta delle Specifiche della realtà di interesse

### Descrizione Sintetica della realtà di interesse

Si vuole progettare una base di dati per una catena di negozi specializzata in vendita di videogiochi, gadget e console.

I vari punti vendita, di cui teniamo traccia del numero di dipendenti che ci lavorano, sono identificati dal loro indirizzo, comprensivo di città e provincia.

In ogni punto vendita sono disponibili per l'acquisto diversi prodotti che rientrano principalmente in 3 categorie: videogiochi, gadget e console. Ogni prodotto è identificato dal proprio codice a barre. Di ogni prodotto ci interessa il nome, il prezzo di listino, la quantità disponibile ed un eventuale sconto; per i gadget salviamo anche il produttore e la serie a cui appartengono.

Per i videogiochi ci interessa sapere la piattaforma, la categoria, la data di rilascio, il numero di giocatori, le condizioni e la valutazione PEGI. Inoltre è disponibile una breve descrizione del prodotto. Per le console vogliamo conservare la famiglia di console e l'anno di rilascio.

Per ogni acquisto effettuato da un cliente vogliamo tener traccia del numero acquisto, che identifica un acquisto, della data in cui è stato effettuato, dell'importo totale.

Vogliamo inoltre tener traccia dei magazzini presso cui i negozi fanno rifornimento. Ci interessa sapere il loro indirizzo, i prodotti al loro interno e la quantità in magazzino.

Per ogni dipendente, identificato dal proprio *employee number*, ci interessa conservare i dati anagrafici, le ore settimanali ed il punto vendita in cui lavorano.

Per titolari soci, di cui ci interessano i dati anagrafici, il punto vendita di cui sono i titolari e la quota percentuale.

Ai clienti è data la possibilità di richiedere una carta fedeltà. Ogni carta fedeltà è legata univocamente ad un cliente, di cui ci interessa conservare i dati anagrafici; di ogni carta ci interessa conservare il livello ed il numero di tessera (che la identifica univocamente).

## Specifiche realtà di interesse

La realtà che andiamo a rappresentare riguarda una catena di negozi specializzata in vendita di videogiochi, gadget e console.

La catena conta 120 punti vendita sparsi per il territorio italiano, per cui è necessario memorizzare l'indirizzo, composto da via, numero civico, città e provincia. Tutti questi dati ci permettono di distinguere in modo univoco i vari punti vendita. Inoltre, ci interessa memorizzare il numero di dipendenti che lavorano in un determinato punto vendita.

Un punto vendita è inserito nel sistema nel momento in cui la sua costruzione è quasi ultimata per cui al momento dell'inserimento non ha ancora dei dipendenti né dei prodotti. Si è però a conoscenza del suo titolare.

I prodotti appartengono a 3 categorie diverse: videogiochi, gadget, console. Ogni prodotto, indipendentemente dalla sua categoria, è identificato in modo univoco dal proprio codice a barre e ci interessa sapere il nome, il prezzo di listino (in euro), la quantità disponibile per l'acquisto in un determinato punto vendita ed un eventuale sconto (in percentuale) applicato al prodotto. Lo sconto è applicato dal singolo punto vendita in caso di eventi o saldi e può quindi essere diverso tra punti vendita diversi. Per il codice a barre utilizziamo lo standard GS1, per cui un codice a barre è composto da 13 cifre che vanno da 0 a 9. In media, si contano 6000 prodotti disponibili per la vendita, di cui 100 sono della categoria "Console", 1500 della categoria "Gadget" ed il rimanente della categoria "Videogiochi".

Per i prodotti che rientrano nella categoria "Videogiochi" vogliamo sapere anche: la piattaforma su cui può essere giocato il videogioco come ad esempio PlayStation, Steam, Nintendo Switch ecc.; le categorie di cui fa parte come Azione, Avventura, Sparatutto in prima persona ecc. (una lista completa delle categorie è disponibile [qui](#)); la data di rilascio (formato gg/mm/aaaa), che può essere una data futura in caso di videogioco disponibile per il preordine; il numero di giocatori, nel formato x-y in cui x indica il numero minimo di giocatori ed y indica il numero massimo di giocatori; le condizioni, nuovo, usato o digitale; una breve descrizione del prodotto (max 250 caratteri); la valutazione PEGI, un metodo di classificazione usato per classificare i videogiochi attraverso cinque categorie di età e otto descrizioni di contenuto. Nello specifico, la categoria di età può assumere uno tra i valori 3, 7, 12, 16, 18; la categoria contenuto può contenere uno o più dei seguenti: Linguaggio scurrile, Discriminazione, Droga, Paura, Gioco d'azzardo, Sesso/Nudità, Violenza, Acquisti integrati, Gioco in rete.

Un videogioco disponibile per più piattaforme può avere un prezzo diverso e/o uno sconto diverso in base alla piattaforma, inoltre il suo codice a barre è diverso in base alla piattaforma.

Per i prodotti che rientrano nella categoria "Gadget" vogliamo sapere anche il produttore e la serie a cui appartengono.

Per i prodotti che rientrano nella categoria “Console” vogliamo sapere anche la famiglia di console di cui fa parte (per esempio la famiglia PlayStation4 include la versione FAT, Slim, Pro ecc.) e l’anno di rilascio, che può essere un anno futuro in caso di console disponibili per il preordine. Fanno parte della categoria console anche i vari accessori quali controller, cuffie ecc. Un prodotto potrebbe essere inserito nel sistema nel momento in cui si aprono i preordini, per cui potrebbe non essere da subito disponibile per l’acquisto. Inoltre, non tutti i prodotti sono disponibili presso ogni punti vendita.

Un cliente può acquistare diversi prodotti sia fisicamente nei punti vendita che tramite l’apposito sito web. Per ogni acquisto, indipendentemente dal tipo, ci interessa memorizzare la data in cui è stato effettuato (formato gg/mm/aaaa), la quantità acquistata di ogni prodotto, l’importo totale ed il metodo di pagamento (Contanti, Visa, MasterCard, PayPal ecc.). Ogni acquisto è identificato dal proprio numero acquisto, un codice numerico il cui formato è xxx-xxxx-xxxx, con ogni x che assume valori compresi tra 0 e 9.

Per gli acquisti online ci interessa inoltre memorizzare l’indirizzo di spedizione e lo stato dell’ordine. Per i preordini vogliamo sapere l’acconto pagato.

Lo stato può assumere i valori “Prenotato”, “Pagamento ricevuto”, “In preparazione alla spedizione”, “Spedito”, “In consegna”, “In custodia” e “Consegnato”. Lo stato “In custodia” è usato in 2 casi: per gli ordini con pagamento in negozio, con cui praticamente “blocchiamo” un determinato prodotto in vendita in un negozio, e per i preordini con consegna in negozio.

Si noti che un acquisto online può anche essere un preordine. In media, si contano 18000 acquisti, di cui 3600 preordini (di cui 2000 sono anche acquisti online) e 7000 acquisti online. Di questi 7000 acquisti online si stimano 3000 acquisti online il cui stato è “In consegna”.

Si noti che i preordini riguardano solo i prodotti della categoria “Videogiochi” ed un preordine contiene solo 1 prodotto al suo interno.

Ogni punto vendita fa rifornimento periodicamente presso uno o più magazzini. Si contano 80 magazzini in totale presso cui viene effettuato il rifornimento. Di ogni magazzino ci interessa sapere l’indirizzo, composto da via, numero civico, città e provincia; dati che ci permettono di identificare univocamente un magazzino. Inoltre ci interessa tener traccia dei vari prodotti presenti in un magazzino e della loro quantità.

La catena conta numerosi dipendenti, ognuno dei quali è identificato dal proprio *employee number*, un valore incrementale composto da 8 cifre. Oltre a questo valore, ci interessa sapere il punto vendita in cui lavorano, le ore settimanali ed i dati anagrafici: nome, cognome, data di nascita (formato gg/mm/aaaa), sesso. In media, ogni punto vendita ha 3 dipendenti. Ogni dipendente è maggiorenne.

Per i titolari soci ci interessa sapere il punto vendita di cui sono i titolari, la quota che posseggono del punto vendita ed i dati anagrafici: nome, cognome, codice fiscale, data di nascita (formato gg/mm/aaaa), sesso. Ogni titolare socio è titolare di un solo punto vendita.

La catena mette a disposizione dei propri clienti una carta fedeltà. Con la carta fedeltà il cliente può accumulare punti per ogni acquisto fatto in modo tale da far salire di livello la carta. Inoltre, con la carta fedeltà si ha accesso a sconti esclusivi (alcuni legati al livello), pre-ordini in anticipo e molto altro. Ogni carta fedeltà è identificata dal proprio numero di tessera, una sequenza di 12 cifre che assumono valori compresi tra 0 e 9. Inoltre, ha un livello indicato da un valore numerico compreso tra 1 e 5 ed è legata in modo univoco ad un cliente. Si contano circa 5000 carte fedeltà. Di ogni cliente ci interessa conservare i dati anagrafici: nome, cognome, data di nascita, sesso, codice fiscale, residenza. Il codice fiscale, che identifica in modo univoco un cliente, è un codice alfanumerico di 16 caratteri. Ogni cliente può essere in possesso di una sola carta fedeltà. Per possedere una carta fedeltà bisogna aver compiuto i 16 anni di età. Si contano 12000 clienti che effettuano acquisti presso i punti vendita della catena o tramite il sito web. Non tutti i clienti sono in possesso di una carta fedeltà.

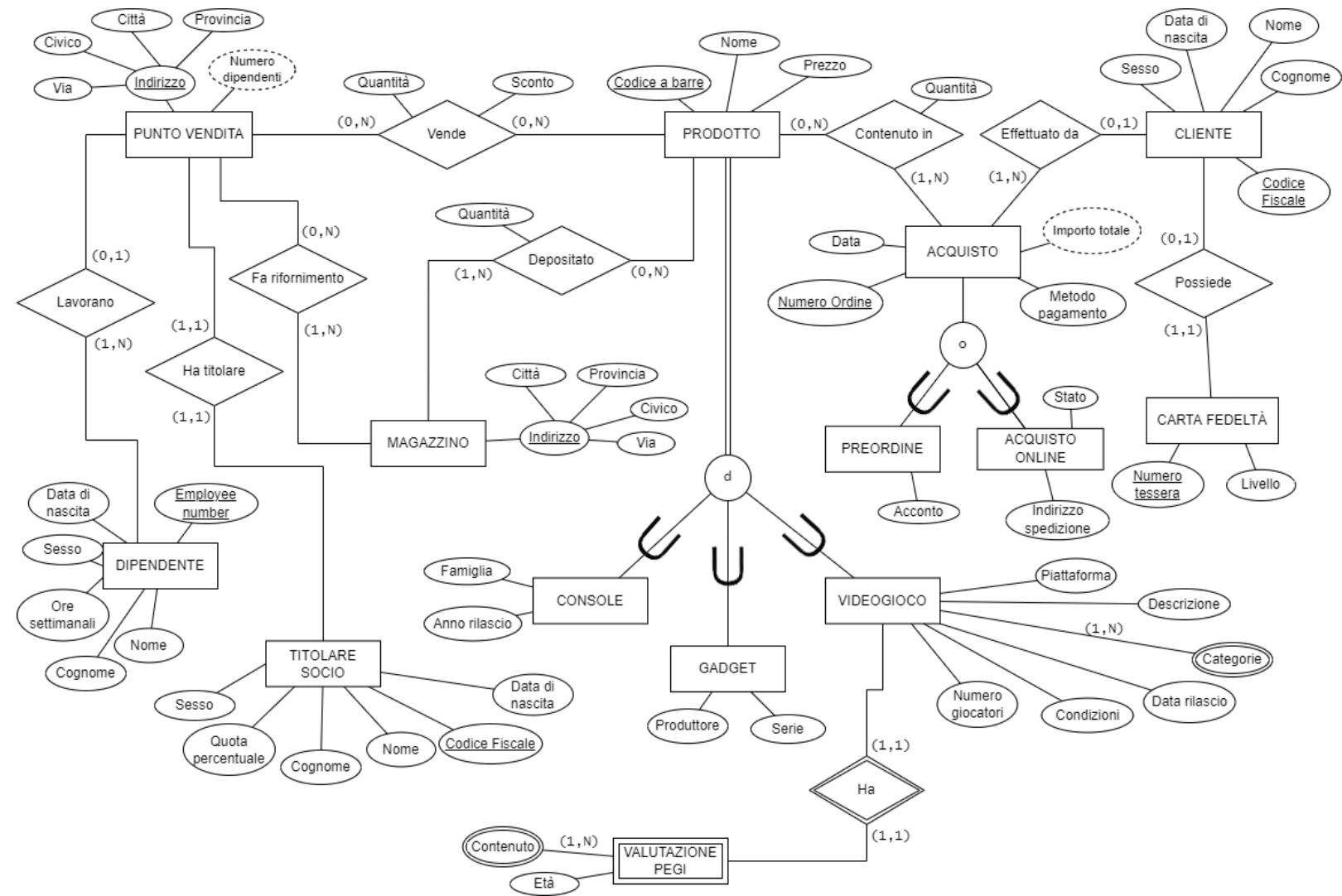
## Glossario dei termini

Termine	Significato
Punto Vendita	Negozio il cui titolare è socio della catena
Prodotto	Prodotto in vendita presso i punti vendita o disponibile a magazzino appartenente a una di tre categorie: videogiochi, gadget e console
Cliente	Persona che effettua acquisti presso i punti vendita o tramite il sito web
Acquisto	Acquisto effettuato presso un negozio della catena oppure tramite il sito web della catena
Dipendente	Persona che lavora presso uno dei punti vendita
Titolare socio	Titolare di uno dei punti vendita della catena
Magazzino	Magazzini presso cui i negozi fanno rifornimento
Carta fedeltà	Carta con cui si raccolgono punti per ogni acquisto effettuato

## Progettazione concettuale della base di dati

### Schema EER

Procedendo con la progettazione concettuale della base di dati, si ottiene il seguente schema EER:



## Dizionario delle Entità

Legenda: **sotto-entità**, **attributo multivалore**, **attributo composto**, **entità debole**, **attributo ridondante**

Entità	Descrizione	Attributi	Identificatore
Punto Vendita	Punto vendita della catena di negozi in cui sono venduti prodotti	<ul style="list-style-type: none"> <li>- <b>Indirizzo:</b> <ul style="list-style-type: none"> <li>- Via</li> <li>- Civico</li> <li>- Città</li> <li>- Provincia</li> </ul> </li> <li>- <b>Numero dipendenti</b></li> </ul>	Indirizzo
Prodotto	Prodotto disponibile per la vendita	<ul style="list-style-type: none"> <li>- Codice a barre</li> <li>- Nome</li> <li>- Prezzo</li> </ul>	Codice a barre
Console	Prodotto disponibile per la vendita appartenente alla categoria “Console”	<ul style="list-style-type: none"> <li>- Famiglia</li> <li>- Anno rilascio</li> </ul>	/
Gadget	Prodotto disponibile per la vendita appartenente alla categoria “Gadget”	<ul style="list-style-type: none"> <li>- Produttore</li> <li>- Serie</li> </ul>	/
Videogioco	Prodotto disponibile per la vendita appartenente alla categoria “Videogioco”	<ul style="list-style-type: none"> <li>- Piattaforma</li> <li>- Descrizione</li> <li>- <b>Categorie</b></li> <li>- Condizioni</li> <li>- Data rilascio</li> <li>- Numero giocatori</li> </ul>	/

Acquisto	Acquisto effettuato da un cliente	<ul style="list-style-type: none"> <li>- Numero Ordine</li> <li>- Data</li> <li>- Metodo pagamento</li> <li>- Importo totale</li> </ul>	Numero Ordine
Preordine	Dati aggiuntivi di un acquisto nel caso questo sia un preordine	<ul style="list-style-type: none"> <li>- Acconto</li> </ul>	/
Acquisto Online	Dati aggiuntivi di un acquisto nel caso questo sia effettuato tramite il sito web	<ul style="list-style-type: none"> <li>- Stato</li> <li>- Indirizzo spedizione</li> </ul>	/
Cliente	Cliente che effettua acquisti presso il negozio	<ul style="list-style-type: none"> <li>- Codice Fiscale</li> <li>- Nome</li> <li>- Cognome</li> <li>- Data di nascita</li> <li>- Sesso</li> </ul>	Codice Fiscale
Carta fedeltà	Tessera fedeltà con cui i clienti possono raccogliere punti	<ul style="list-style-type: none"> <li>- Numero tessera</li> <li>- Livello</li> </ul>	Numero tessera
Valutazione PEGI	Valutazione di un videogioco in base ai contenuti	<ul style="list-style-type: none"> <li>- Età</li> <li>- Contenuto</li> </ul>	/
Magazzino	Magazzino presso cui i punti vendita fanno rifornimento	<ul style="list-style-type: none"> <li>- Indirizzo: <ul style="list-style-type: none"> <li>- Via</li> <li>- Civico</li> <li>- Città</li> <li>- Provincia</li> </ul> </li> </ul>	Indirizzo

Dipendente	Dipendente che lavora presso uno dei punti vendita	<ul style="list-style-type: none"> <li>- Employee number</li> <li>- Nome</li> <li>- Cognome</li> <li>- Data di nascita</li> <li>- Sesso</li> <li>- Ore settimanali</li> </ul>	Employee Number
Titolare Socio	Titolare di uno dei punti vendita	<ul style="list-style-type: none"> <li>- Codice Fiscale</li> <li>- Nome</li> <li>- Cognome</li> <li>- Data di nascita</li> <li>- Sesso</li> <li>- Quota percentuale</li> </ul>	Codice Fiscale

## Dizionario delle Relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Vende	Un negozio vende i suoi prodotti	Punto vendita (0,N) Prodotto (0,N)	<ul style="list-style-type: none"> <li>- Quantità</li> <li>- Sconto</li> </ul>
Contenuto in	Un prodotto è contenuto in un acquisto	Prodotto (0,N) Acquisto (1,N)	<ul style="list-style-type: none"> <li>- Quantità</li> </ul>
Effettuato da	Un acquisto è effettuato da un cliente	Acquisto (1,N) Cliente (0,1)	/
Possiede	Un cliente possiede una carta fedeltà	Cliente (0,1) Carta fedeltà (1,1)	/

Ha	Un videogioco ha una valutazione PEGI	Videogioco (1,1) Valutazione PEGI (1,1)	/
Depositato	Un prodotto è depositato presso un magazzino	Prodotto (0,N) Magazzino (1,N)	/
Fa rifornimento	Un punto vendita fa rifornimento presso un magazzino	Punto vendita (0,N) Magazzino (1,N)	/
Lavorano	Un dipendente lavora presso un punto vendita	Punto Vendita (0,1) Dipendente (1,N)	/
Ha titolare	Un punto vendita ha un titolare	Punto Vendita (1,1) Titolare socio (1,1)	/

### Vincoli non esprimibili nello schema

Valgono i seguenti vincoli di dominio:

- Tutte le date sono in formato aaaa-mm-gg
- Nell'entità Prodotto, l'attributo codice a barre è una sequenza di 13 cifre, che assumono valori compresi tra 0 e 9
- Nell'entità Console, l'attributo anno di rilascio può assumere il valore di una data futura nel caso di console disponibili per il preordine
- Nell'entità Videogioco, l'attributo data di rilascio può assumere il valore di una data futura nel caso di videogiochi disponibili per il preordine
- Nell'entità Videogioco, l'attributo Descrizione ha dimensione massima di 250 caratteri
- Nell'entità Videogioco, l'attributo Categorie può assumere uno o più valori presi da [questa lista](#)

- Nell'entità Videogioco, l'attributo Numero giocatori ha formato x-y, in cui x indica il numero minimo di giocatori ed y indica il numero massimo di giocatori
- Nell'entità Acquisto, l'attributo Numero Ordine è una sequenza di numeri nel formato xxx-xxxx-xxxx, ogni x è una cifra il cui valore è compreso tra 0 e 9
- Nell'entità Acquisto, l'attributo Stato assume uno solo dei seguenti valori: Prenotato, Pagamento ricevuto, In preparazione alla spedizione, Spedito, In Consegnna, In Custodia, Consegnato
- Nell'entità Cliente, l'attributo Codice Fiscale è composto da 16 caratteri, ogni carattere può essere una cifra compresa tra 0 e 9 oppure una lettera dell'alfabeto (in maiuscolo)
- Nell'entità Cartà Fedeltà, l'attributo Numero tessera è una sequenza di 12 cifre, che assumono valori compresi tra 0 e 9
- Nell'entità Cartà Fedeltà, l'attributo livello è un numero compreso tra 1 e 5
- Nell'entità Valutazione PEGI, l'attributo Età può assumere uno solo tra i seguenti valori: 3, 7, 12, 16, 18
- Nell'entità Valutazione PEGI, l'attributo Categorie può assumere uno o più valori tra: Linguaggio scurrile, Discriminazione, Droga, Paura, Gioco d'azzardo, Sesso/Nudità, Violenza, Acquisti integrati, Gioco in rete
- Nell'entità Dipendente, l'attributo employee number è una sequenza di 8 cifre, che assumono valori compresi tra 0 e 9

Valgono le seguenti regole di vincolo:

- Un cliente inserito nel sistema deve avere almeno 16 anni compiuti
- Un dipendente deve avere almeno 18 anni compiuti

## Definizione delle procedure per la gestione della base di dati

Tavola dei Volumi

Concetto	Tipo	Carico applicativo
Punto Vendita	E	120
Prodotto	E	6000
Console	E	100
Gadget	E	1500
Videogioco	E	4400
Acquisto	E	18000
Preordine	E	3600
Acquisto online	E	7000
Cliente	E	12000
Carta fedeltà	E	5000
Valutazione PEGI	E	4400
Magazzino	E	80
Dipendente	E	360
Titolare Socio	E	120
Vende	R	700000
Contenuto in	R	54000
Effettuato da	R	18000
Possiede	R	5000

Ha	R	4400
Depositato	R	6000
Fa rifornimento	R	120
Lavora	R	360
Ha titolare	R	120

### Tavola delle Operazioni

	Operazione	Tipo	Frequenza
1	Inserire un nuovo videogioco	I	25/settimana
2	Inserire un nuovo gadget	I	15/mese
3	Inserire una nuova console	I	6/anno
4	Inserire un nuovo dipendente	I	10/anno
5	Inserire un nuovo punto vendita	I	2/anno
6	Inserire una carta fedeltà	I	20/settimana
7	Inserire un cliente	I	20/settimana
8	Elencare tutti i clienti in possesso di una carta fedeltà almeno di livello 3	I	5/settimana
9	Elencare tutte le informazioni sui videogiochi la cui valutazione PEGI di età è 18	I	70/settimana

14

10	Elencare tutti i dipendenti che lavorano presso un determinato punto vendita	I	10/mese
11	Creare un elenco di tutti i punti vendita con nome e cognome dei rispettivi titolari soci	B	5/anno
12	Elencare tutte le informazioni sugli acquisti effettuati da un determinato cliente, assieme al nome e alla quantità dei vari prodotti acquistati per ogni acquisto	I	6000/settimana
13	Elencare la disponibilità totale per ogni prodotto, che includa sia la disponibilità nei punti vendita che quella disponibili in magazzini	I	700/settimana
14	Creare un elenco dei prodotti la cui disponibilità presso un determinato punto vendita è 0	I	300/settimana
15	Elencare tutti i videogiochi rilasciati in un determinato anno	I	150/settimana
16	Creare un elenco che conteggia i gadget in base alla serie in ordine decrescente	I	20/mese
17	Elencare tutti i prodotti in vendita presso un determinato punto vendita	I	700/settimana
18	Elencare tutte le console appartenenti alla stessa famiglia	I	100/mese
19	Elencare tutte le informazioni riguardanti i preordini, includendo tutte le informazioni sui videogiochi	I	5000/settimana

20	Elencare tutti i prodotti in sconto disponibili presso un punto vendita	I	1000/settimana
21	Elencare tutte le informazioni sugli ordini in consegna	I	1000/settimana

## Progettazione logica

### Analisi delle ridondanze

Le uniche ridondanze presenti nel modello sono l'attributo “Numero dipendenti” in “Punto vendita” e l'attributo “Importo totale” in Acquisto.

Per quanto riguarda l'attributo “Numero dipendenti”, le operazioni che lo coinvolgono sono la 4, la 5 e la 11.

Tutte e 3 le operazioni hanno bassissima frequenza rispetto alle altre: sono le uniche operazioni (oltre l'operazione 3) ad avere frequenza annua, rispettivamente 10 volte l'anno, 2 volte l'anno e 5 volte l'anno. Inoltre, l'operazione 11 è eseguita in modalità batch. Inoltre, dall'analisi dei dati risulta che il carico applicativo dell'entità Punto Vendita è pari a 120, per cui assumendo 2 byte per ogni attributo “Numero dipendenti”, avremo  $2 \times 120 = 240$  byte di spazio ridondante.

Pertanto, non è ritenuta necessaria l'eliminazione dell'attributo ridondante “Numero dipendenti” in quanto l'impatto sulle performance è trascurabile.

Per quanto riguarda l'attributo “Importo totale”, le operazioni che lo coinvolgono sono la 12, la 19 e la 21.

Supponendo che un attributo occupi 2 byte, sarebbero necessari  $2 \times 18000 = 36000$  byte = 36 kb di memoria per il dato ridondante. E' necessario valutare il numero di accessi per sapere se vale la pena o meno questo spreco di memoria.

Di seguito, riportiamo le tavole degli accessi per le varie operazioni, suddividendole per presenza o assenza del dato ridondante.

Per semplicità, assegnamo un costo pari ad 1 per ogni operazione di lettura (L).

Operazione 12 (6000 volte/settimana)							
Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Cliente	E	1	L	Cliente	E	1	L
Acquisto	E	2	L	Acquisto	E	2	L
Acquisto online	E	1	L	Acquisto online	E	1	L
Contenuto in	R	6	L	Contenuto in	R	6	L
Prodotto	E	6	L	Prodotto	E	6	L
<b>Totale:</b> 16L x 6000 = 96000 accessi/settimana				<b>Totale:</b> 16L x 6000 = 96000 accessi/settimana			

Dalla tavola dei volumi deduciamo che, in media, ogni acquisto contiene in media 3 prodotti (54000 Contenuto in / 18000 Acquisto) e ogni cliente effettua 2 acquisti (18000 Effettuato da/ 12000 Cliente, arrotondiamo per eccesso).

Inoltre, dato che circa la metà degli acquisti è anche un acquisto online oppure un preordine, si fa l'assunzione che uno dei due acquisti effettuati in media dai clienti richieda l'accesso ad una tra Preordine ed Acquisto online. Nella tabella è riportato l'esempio con accesso all'entità Acquisto Online. Sia noto, però, che nel caso in cui uno dei due ordini sia un preordine sono richiesti meno accessi alla relazione Contenuto in ed all'entità Prodotto in quanto, come espresso nelle specifiche, un preordine riguarda solo un prodotto.

Operazione 19 (5000 volte/settimana)							
Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Acquisto	E	3600	L	Acquisto	E	3600	L
Preordine	E	3600	L	Preordine	E	3600	L
Acquisto online	E	2000	L	Acquisto online	E	2000	L
Contenuto in	R	3600	L	Contenuto in	R	3600	L
Prodotto	E	3600	L	Prodotto	E	3600	L
Videogioco	E	3600	L	Videogioco	E	3600	L
Valutazione PEGI	E	3600	L	Valutazione PEGI	E	3600	L
<b>Totale:</b> (3600L x 6 + 2000L) x 5000 = 118M accessi/settimana				<b>Totale:</b> (3600L x 6 + 2000L) x 5000 = 118M accessi/settimana			

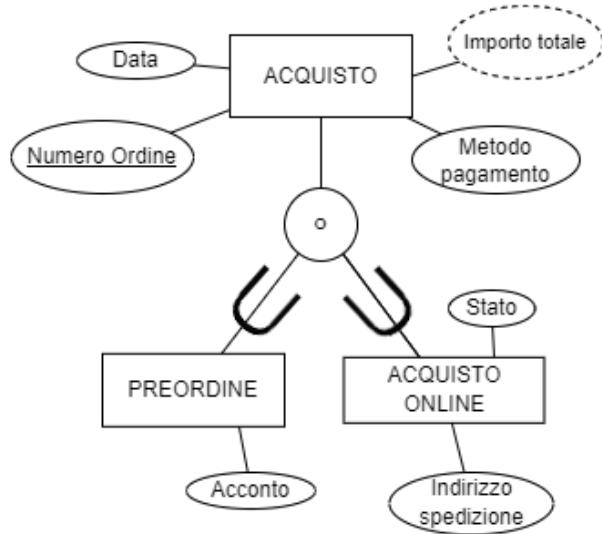
Operazione 21 (1000 volte/settimana)							
Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Acquisto	E	3000	L	Acquisto	E	3000	L
Acquisto online	E	3000	L	Acquisto online	E	3000	L
				Contenuto in	R	9000	L
				Prodotto	E	9000	L
<b>Totale:</b> 6000L x 1000 = 6M accessi/settimana				<b>Totale:</b> 24000L x 1000 = 24M accessi/settimana			

Totale accessi			
Calcolo con ridondanza		Calcolo senza ridondanza	
Operazione	Accessi/settimana	Operazione	Accessi/settimana
12	96000	12	96000
19	118M	19	118M
21	6M	21	24M
<b>Totale:</b> 124.096.000 accessi/settimana		<b>Totale:</b> 142.096.000 accessi/settimana	

Visto il notevole minor numero di accessi su operazioni molto frequenti ed un costo in memoria relativamente basso (35kb) manteniamo l'attributo ridondante.

### Eliminazione delle gerarchie

Sono presenti 2 generalizzazioni: quella di Acquisto e quella di Prodotto.  
Analizziamo prima la generalizzazione "Acquisto".



La generalizzazione non è totale e presenta delle sovrapposizioni.

Inoltre le figlie presentano attributi che le contraddistinguono.

Le operazioni che coinvolgono la generalizzazione sono la 12, la 19 e la 21.

Valutiamo tutte le opzioni disponibili:

- Accorpando il padre nelle figlie andremo a perdere i dati di tutti gli acquisti che non sono né preordini né acquisti online, per cui questa opzione è da scartare a priori
- Accorpando le figlie nel padre riduciamo il numero di accessi per le operazioni ma introduciamo valori nulli
- Rendendo le figlie delle entità deboli e creando delle relazioni con esse manteniamo lo stesso numero di accessi ma non creiamo dei valori nulli

Dato che le operazioni che riguardano la generalizzazione sono operazioni che richiedono un numero elevatissimo di accessi (poco più di 124 milioni a settimana, come visto nel calcolo della ridondanza) calcoliamo il numero di valori nulli introdotti ed il numero di accessi per valutare se conviene accorpate o meno. Riporteremo solo le tabelle per gli accessi in caso di accorpamento dato che le tabelle degli accessi senza accorpamento sono già presenti nel paragrafo delle analisi delle ridondanze.

In caso andassimo ad accorpare, ci sarebbero all'incirca 33600 valori nulli.

Inoltre le tabelle degli accessi per le varie operazioni sono le seguenti:

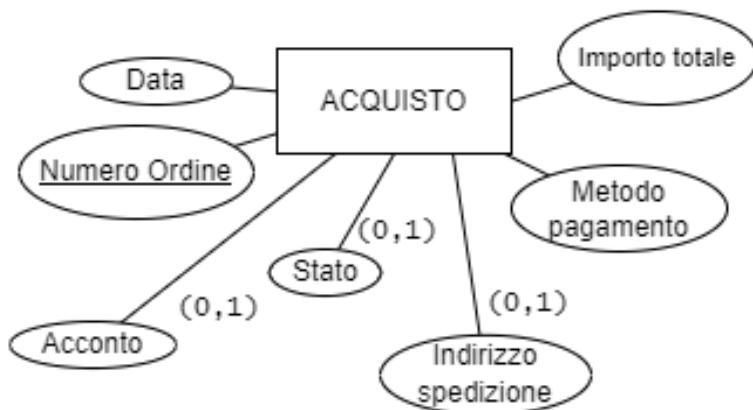
Operazione 12 (6000 volte/settimana)				Operazione 19 (5000 volte/settimana)			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Cliente	E	1	L	Acquisto	E	3600	L
Acquisto	E	2	L	Contenuto in	R	3600	L
Contenuto in	R	6	L	Prodotto	E	3600	L
Prodotto	E	6	L	Videogioco	E	3600	L
				Valutazione PEGI	E	3600	L
<b>Totale: 15L x 6000 = 90000 accessi/settimana</b>				<b>Totale: (3600L x 5) x 5000 = 90M accessi/settimana</b>			

Operazione 21 (1000 volte/settimana)			
Tabella	Tipo	Accessi	Tipo Accessi
Acquisto	E	3000	L
<b>Totale:</b> 3000L x 1000 = 3M accessi/settimana			

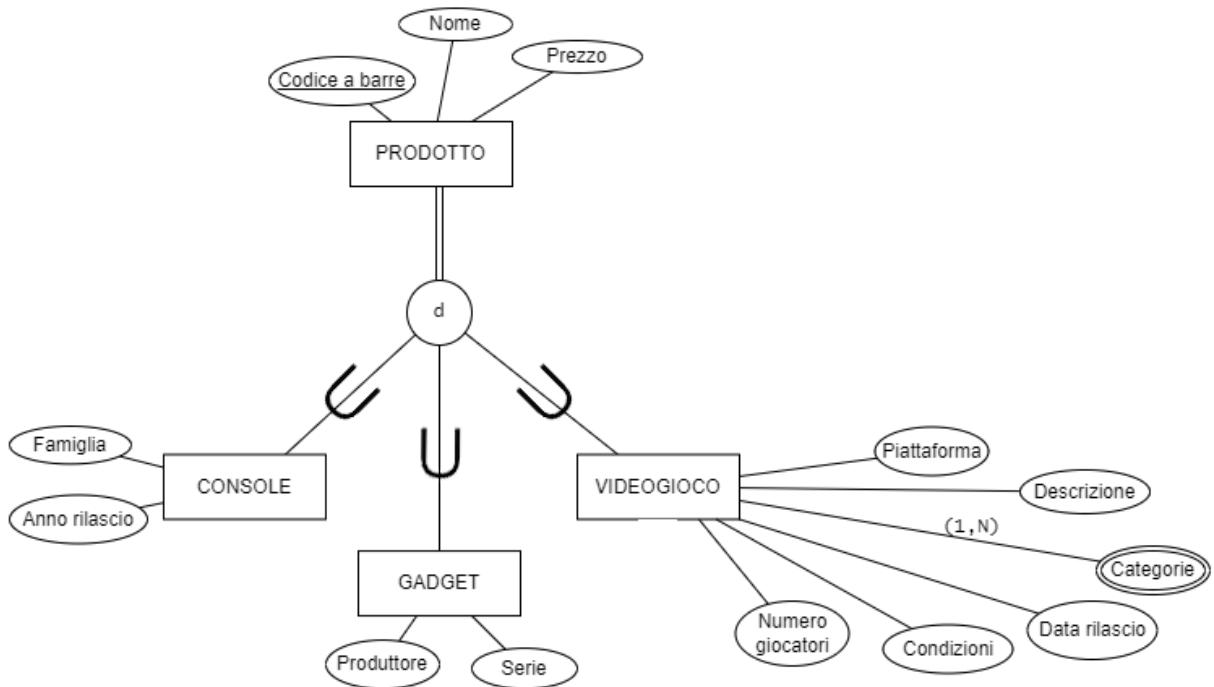
Totale accessi			
Calcolo senza accorpamento		Calcolo con accorpamento	
Operazione	Accessi/settimana	Operazione	Accessi/settimana
12	96000	12	90000
19	118M	19	90M
21	6M	21	3M
<b>Totale:</b> 124.096.000 accessi/settimana		<b>Totale:</b> 93.090.000 accessi/settimana	

In caso di accorpamento, vi è un'enorme riduzione degli accessi (31M di accessi in meno a settimana!!) ma vi è anche l'introduzione di parecchi valori nulli (33600).

Visto che si tratta di operazioni molto frequenti sceglieremo l'ottimizzazione ed il minor numero di accessi, per cui accorpiamo le entità figlie nell'entità padre.



Valutiamo la generalizzazione “Prodotto”



In questo caso la generalizzazione è totale e non presenta sovrapposizioni.

Inoltre, le entità figlie hanno attributi che le distinguono, numerosi nel caso di “Videogioco” che ha anche una relazione con un’entità debole.

Le operazioni che coinvolgono l’entità Prodotto e le figlie sono numerose, le dinguiamo in:

- operazioni che fanno distinzione tra le figlie: 1, 2, 3, 15, 16, 18, 19 (totale: 5175 volte a settimana + 135 volte/mese + 6 volte/anno)
- operazioni che non fanno distinzione tra le figlie (totale: 8700 volte/settimana)

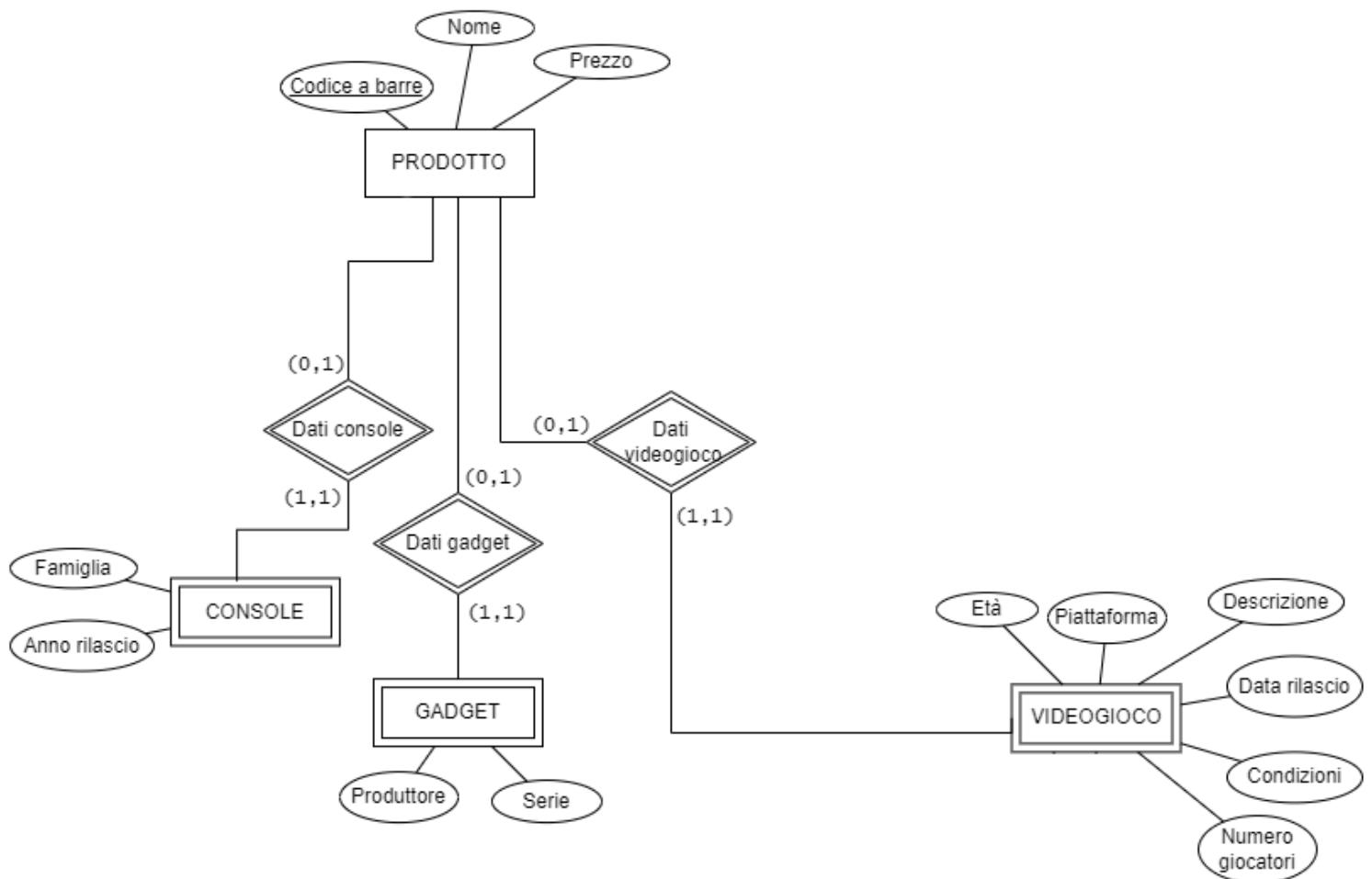
Valutiamo le varie opzioni disponibili:

- L’accorpamento delle figlie nel padre creerebbe molti valori nulli per via dei molti attributi dell’entità figlia Videogioco (numeri??), per cui scartiamo questa opzione.
- L’accorpamento del padre nelle figlie ridurrebbe il numero di accesso per le operazioni che fanno distinzioni tra le figlie, ma le relazioni “Vende”, “Depositato” e “Contenuto in” dovrebbero essere triplicate per rispecchiare questo cambiamento

- Rendendo le figlie delle entità deboli e creando delle relazioni con esse manteniamo lo stesso numero di accessi ma non creiamo dei valori nulli

Scartiamo la scelta dell'accorpamento delle padri nelle figlie in quanto vi è un aumento nel numero di relazioni visitate, e questo rallenta le operazioni. Inoltre, il gruppo di operazioni che ne risentirebbe di più è quello con una frequenza settimanale maggiore di quello che ne beneficierebbe dall'accorpamento.

Quindi, decidiamo di mantenere la suddivisione tra padre e figlie rendendo le figlie delle entità deboli e creando delle relazioni con l'entità padre padre. Ogni entità figlia è identificata dalla chiave primaria dell'entità padre Prodotto".



## Partizionamento/Accorpamento dei concetti

L'entità debole Valutazione PEGI contiene dati aggiuntivi su un Videogioco che vengono richiesti in ogni operazione in cui è presente l'entità debole Videogioco.

Inoltre, vi è una relazione 1-a-1 tra le due con partecipazione totale di entrambe.

Per cui, accorpiamo l'entità debole Valutazione PEGI nell'entità debole Videogioco, riducendo il numero di accessi..

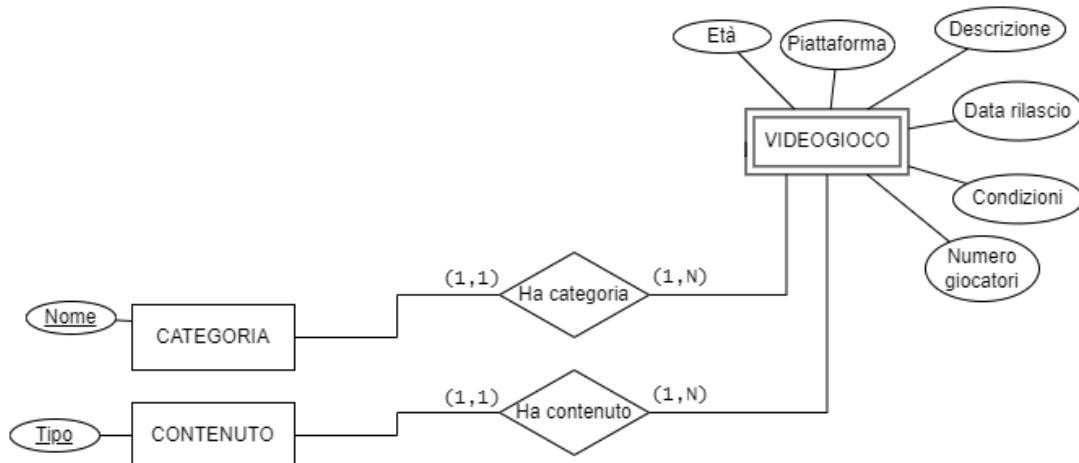


Vanno rimossi gli attributi multivalue dalle entità.

Ne sono presenti 2: Categorie e Contenuto, entrambi in Videogioco.

Trasformiamo questi attributi in due entità deboli, creando delle relazioni 1-a-N tra Videogioco e queste nuove entità.

Rispettivamente, Categoria è identificato dal Nome e dalla chiave primaria di videogioco, e Contenuto è identificato dal Tipo e dalla chiave primaria del videogioco.

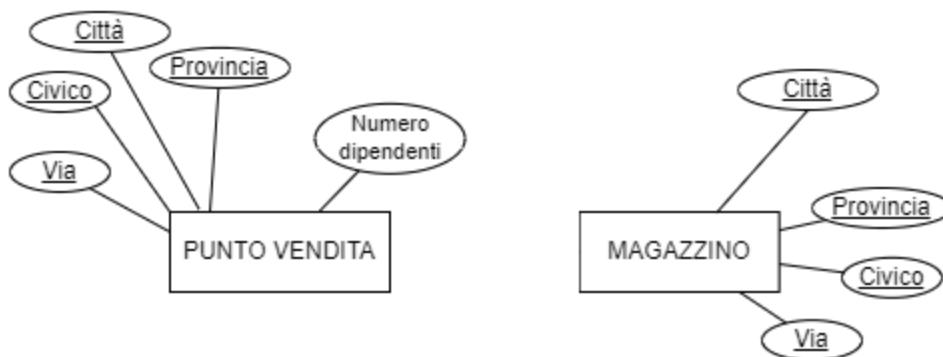


Decidiamo, infine, di rimuovere gli attributi.

Sono presenti 2 attributi composti, rispettivamente nelle entità Punto Vendita e Magazzino.

In entrambi i casi l'attributo si chiama Indirizzo ed è composto da: Via, Civico, Città, Provincia.

Eliminiamo l'attributo composto Indirizzo ed inseriamo gli attributi che lo componevano in entrambe le entità.



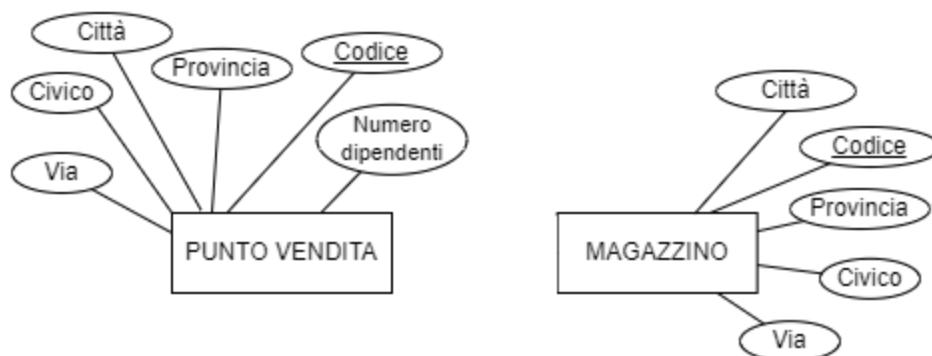
### Scelta identificatori primari

Tutte le entità hanno un solo identificatore primario, che già in fase di progettazione concettuale sono stati scelti come identificatori per le entità.

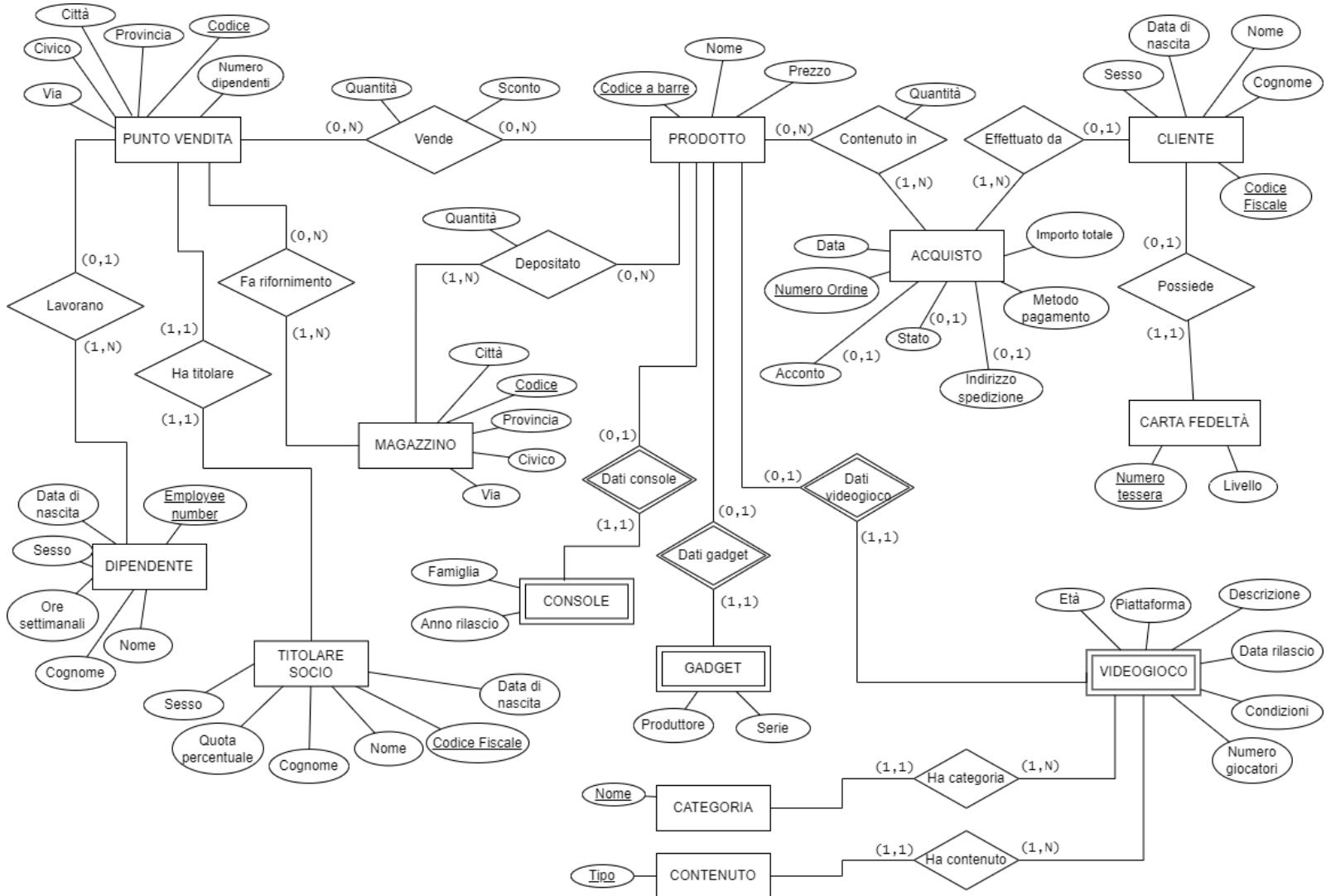
Questi identificatori diventano quindi le chiavi primarie delle rispettive entità.

Va fatta, però, una considerazione sulle entità Punto Vendita e Magazzino.

La chiave primaria di entrambe queste entità è composta dagli attributi Via, Civico, Città, Provincia. Una chiave di questo genere risulta molto pesante per lo svolgimento delle operazioni di ricerca, perciò assegnamo ad entrambe le entità un codice numerico incrementale come chiave primaria.



## Schema ER ristrutturato



## Schema relazionale

Procediamo quindi al mapping della base di dati:

**PuntoVendita**(codice, via, civico, citta, provincia, numeroDipendenti)

**Dipendente**(employeeNumber, nome, cognome, dataDiNascita, sesso, oreSettimanali, PuntoVendita.codice↑)

**TitolareSocio**(codiceFiscale, nome, cognome, dataDiNascita, sesso, quota, PuntoVendita.codice↑)

**Magazzino**(codice, via, civico, citta, provincia)

**Prodotto**(barcode, nome, prezzo)

**Console**(Prodotto.barcode↑, famiglia, annoRilascio)

**Gadget**(Prodotto.barcode↑, produttore, serie)

**Videogioco**(Prodotto.barcode↑, piattaforma, descrizione, dataRilascio, condizioni, numeroGiocatori, etaPEGI)

**Categoria**(nome, Videogioco.Prodotto.barcode↑)

**Contenuto**(tipo, Videogioco.Prodotto.barcode↑)

**Acquisto**(numeroOrdine, data, metodoPagamento, importoTotale, acconto\*, stato\*, indirizzo Spedizione\*)

**Cliente**(codiceFiscale, nome, cognome, dataDiNascita, sesso)

**CartaFedelta**(numeroTessera, livello, Cliente.codiceFiscale↑)

**FA\_RIFORNIMENTO**(PuntoVendita.codice↑, Magazzino.codice↑)

**VENDE**(PuntoVendita.codice↑, Prodotto.barcode↑, quantita, sconto)

**DEPOSITATO**(Prodotto.barcode↑, Magazzino.codice↑, quantita)

**CONTENUTO\_IN**(Prodotto.barcode↑, Acquisto.numeroOrdine↑, quantita)

**EFFETTUATO\_DA**(Acquisto.numeroOrdine↑, Cliente.codiceFiscale↑)

## Normalizzazione

Lo schema si presenta già normalizzato.

Lo schema è già in prima forma normale in quanto, già in fase di ristrutturazione, abbiamo eliminato attributi multivалore e composti, quindi ogni attributo nella relazione è atomico.

Lo schema è in seconda forma normale perché, oltre ad essere in prima forma normale, quando è presente una chiave primaria composta da più attributi, tutte le dipendenze funzionali che la riguardano sono piene e non parziali.

Lo schema è in terza forma normale perché, oltre ad essere in seconda forma normale, in tutte le tabelle non sono presenti dipendenze funzionali transitive fra attributi non chiave e la chiave primaria.

## Realizzazione della basi di dati con MySQL

Di seguito il listato delle istruzioni MySQL per la creazione della base di dati:

```

1 •  drop database if exists CatenaNegozi;
2 •  create database CatenaNegozi;
3 •  use CatenaNegozi;
4 •  drop user 'negozi'@'%';
5 •  create user 'negozi'@'%' identified by 'negozi';
6 •  grant all on negozi.* to 'negozi'@'%';

8 •  drop table if exists PuntoVendita;
9 •  create table PuntoVendita
10 (
11     codice int PRIMARY KEY NOT NULL auto_increment,
12     via varchar(50) NOT NULL,
13     civico varchar(3) NOT NULL,
14     citta varchar(30) NOT NULL,
15     provincia char(2) NOT NULL,
16     numeroDipendenti int NOT NULL
17 );

```

```
19 • drop table if exists Dipendente;
20 • create table Dipendente
21   (
22     employeeNumber char(8) PRIMARY KEY NOT NULL,
23     nome varchar(20) NOT NULL,
24     cognome varchar(20) NOT NULL,
25     dataDiNascita date NOT NULL,
26     sesso char(1) NOT NULL,
27     oreSettimanali int NOT NULL,
28     puntoVendita int NOT NULL,
29     FOREIGN KEY (puntoVendita) REFERENCES PuntoVendita(codice) ON DELETE CASCADE ON UPDATE CASCADE,
30     CHECK (2022 - YEAR(dataDiNascita) >= 18) -- vincolo sull'età dei dipendenti --
31   );
32
33 • drop table if exists TitolareSocio;
34 • create table TitolareSocio
35   (
36     codiceFiscale char(16) PRIMARY KEY NOT NULL,
37     nome varchar(20) NOT NULL,
38     cognome varchar(20) NOT NULL,
39     dataDiNascita date NOT NULL,
40     sesso char(1) NOT NULL,
41     quota int NOT NULL,
42     puntoVendita int NOT NULL,
43     FOREIGN KEY (puntoVendita) REFERENCES PuntoVendita(codice) ON DELETE CASCADE ON UPDATE CASCADE
44   );
45
46 • drop table if exists Magazzino;
47 • create table Magazzino
48   (
49     codice int PRIMARY KEY NOT NULL auto_increment,
50     via varchar(50) NOT NULL,
51     civico varchar(3) NOT NULL,
52     citta varchar(30) NOT NULL,
53     provincia char(2) NOT NULL
54   );
55
56 • drop table if exists Prodotto;
57 • create table Prodotto
58   (
59     barcode char(13) PRIMARY KEY NOT NULL,
60     nome varchar(50) NOT NULL,
61     prezzo float NOT NULL
62   );
```

```
64 • drop table if exists Console;
65 • create table Console
66   (
67     prodotto char(13) PRIMARY KEY NOT NULL,
68     famiglia varchar(30) NOT NULL,
69     annoRilascio int NOT NULL,
70     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE
71   );
72
73 • drop table if exists Gadget;
74 • create table Gadget
75   (
76     prodotto char(13) PRIMARY KEY NOT NULL,
77     produttore varchar(30) NOT NULL,
78     serie varchar(50) NOT NULL,
79     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE
80   );
82 • drop table if exists Videogioco;
83 • create table Videogioco
84   (
85     prodotto char(13) NOT NULL PRIMARY KEY,
86     piattaforma varchar(20) NOT NULL,
87     descrizione varchar(250) NOT NULL,
88     dataRilascio date NOT NULL,
89     condizioni varchar(10) NOT NULL,
90     numeroGiocatori char(3) NOT NULL,
91     etaPEGI int NOT NULL,
92     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE
93   );
```

```
95 • drop table if exists Categoria;
96 • create table Categoria
97 (
98     nome varchar(30) NOT NULL,
99     videogioco char(13) NOT NULL,
100    PRIMARY KEY(nome, videogioco),
101    FOREIGN KEY (videogioco) REFERENCES Videogioco(prodotto) ON DELETE CASCADE ON UPDATE CASCADE
102 );
103
104 • drop table if exists Contenuto;
105 • create table Contenuto
106 (
107     tipo varchar(20) NOT NULL,
108     videogioco char(13) NOT NULL,
109     PRIMARY KEY(tipo, videogioco),
110     FOREIGN KEY (videogioco) REFERENCES Videogioco(prodotto) ON DELETE CASCADE ON UPDATE CASCADE
111 );
112
113 • drop table if exists Acquisto;
114 • create table Acquisto
115 (
116     numeroOrdine char(13) PRIMARY KEY NOT NULL,
117     dataAcquisto date NOT NULL,
118     metodoPagamento varchar(15) NOT NULL,
119     importoTotale float NOT NULL,
120     acconto float,
121     stato varchar(30),
122     indirizzoSpedizione varchar(150)
123 );
```

```
125 • drop table if exists Cliente;
126 • create table Cliente
127   (
128     codiceFiscale char(16) PRIMARY KEY NOT NULL,
129     nome varchar(20) NOT NULL,
130     cognome varchar(30) NOT NULL,
131     sesso char(1) NOT NULL,
132     dataDiNascita date NOT NULL
133     CHECK (2022 - YEAR(dataDiNascita) >= 16) -- vincolo sull'età dei clienti --
134   );
135
136 • drop table if exists CartaFedelta;
137 • create table CartaFedelta
138   (
139     numeroTessera char(12) PRIMARY KEY NOT NULL,
140     livello int NOT NULL,
141     cliente char(16) NOT NULL,
142     FOREIGN KEY (cliente) REFERENCES Cliente(codiceFiscale) ON DELETE CASCADE ON UPDATE CASCADE
143   );
144
145 • drop table if exists FA_RIFORNIMENTO;
146 • create table FA_RIFORNIMENTO
147   (
148     puntoVendita int NOT NULL,
149     magazzino int NOT NULL,
150     PRIMARY KEY(puntoVendita, magazzino),
151     FOREIGN KEY (puntoVendita) REFERENCES PuntoVendita(codice) ON DELETE CASCADE ON UPDATE CASCADE,
152     FOREIGN KEY (magazzino) REFERENCES Magazzino(codice) ON DELETE CASCADE ON UPDATE CASCADE
153   );
154
155 • drop table if exists VENDE;
156 • create table VENDE
157   (
158     puntoVendita int NOT NULL,
159     prodotto char(13) NOT NULL,
160     quantita int NOT NULL,
161     sconto int NOT NULL,
162     PRIMARY KEY(puntoVendita, prodotto),
163     FOREIGN KEY (puntoVendita) REFERENCES PuntoVendita(codice) ON DELETE CASCADE ON UPDATE CASCADE,
164     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE
165   );
```

```
167 • drop table if exists DEPOSITATO;
168 • create table DEPOSITATO
169   (
170     prodotto char(13) NOT NULL,
171     magazzino int NOT NULL,
172     quantita int NOT NULL,
173     PRIMARY KEY(prodotto, magazzino),
174     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE,
175     FOREIGN KEY (magazzino) REFERENCES Magazzino(codice) ON DELETE CASCADE ON UPDATE CASCADE
176   );
177
178 • drop table if exists CONTENUTO_IN;
179 • create table CONTENUTO_IN
180   (
181     prodotto char(13) NOT NULL,
182     acquisto char(13) NOT NULL,
183     quantita int not null,
184     PRIMARY KEY (prodotto, acquisto),
185     FOREIGN KEY (prodotto) REFERENCES Prodotto(barcode) ON DELETE CASCADE ON UPDATE CASCADE,
186     FOREIGN KEY (acquisto) REFERENCES Acquisto(numeroOrdine) ON DELETE CASCADE ON UPDATE CASCADE
187   );
188
189
190
191
192
193
```

## Implementazione query SQL

### Operazione 1:

Vanno inseriti i dati in: prodotto, videogioco, categoria e contenuto

Inserimento in prodotto:

```
INSERT INTO prodotto(barcode, nome, prezzo) VALUES (?,?,?);
```

Inserimento in videogioco:

```
INSERT INTO videogioco(prodotto, piattaforma, descrizione, dataRilascio, condizioni, numeroGiocatori, etaPEGI) VALUES (?,?,?,?,?,?,?);
```

Inserimento in categoria:

```
INSERT INTO categoria(nome, videogioco) VALUES (?,?);
```

Inserimento in contenuto:

```
INSERT INTO contenuto(tipo, videogioco) VALUES (?,?);
```

### Operazione 2:

Vanno inseriti i dati in prodotto e gadget

Inserimento in prodotto:

```
INSERT INTO prodotto(barcode, nome, prezzo) VALUES (?,?,?);
```

Inserimento in gadget:

```
INSERT INTO gadget(prodotto, produttore, serie) VALUES (?,?,?);
```

**Operazione 3:**

Vanno inseriti i dati in prodotto e console

Inserimento in prodotto:

```
INSERT INTO prodotto(barcode, nome, prezzo) VALUES (?,?,?);
```

Inserimento in console:

```
INSERT INTO console(prodotto, famiglia, annoRilascio) VALUES (?,?,?);
```

**Operazione 4:**

```
INSERT INTO dipendente(employeeNumber, nome, cognome, dataDiNascita, sesso, oreSettimanali, puntoVendita)
VALUES (?,?,?,?,?,?);
```

**Operazione 5:**

```
INSERT INTO puntoVendita(codice, via, civico, citta, provincia, numeroDipendenti)
VALUES (?,?,?,?,?,?);
```

**Operazione 6:**

```
INSERT INTO cartaFedelta(numeroTessera, livello, cliente) VALUES (?,?,?);
```

**Operazione 7:**

```
INSERT INTO cliente(codiceFiscale, nome, cognome, dataDiNascita, sesso) VALUES (?,?,?,?,?);
```

**Operazione 8:**

```
SELECT c.* FROM cliente c, cartafedelta cf WHERE c.codiceFiscale = cf.codiceFiscale AND
cf.livello = ?;
```

**Operazione 9:**

```
SELECT * FROM videogioco v, prodotto p, categoria c, contenuto cnt  
WHERE v.etaPEGI = 18 AND v.prodotto = p.barcode AND v.prodotto = c.videogioco AND  
v.videogioco = cnt.videogioco  
GROUP BY barcode
```

**Operazione 10:**

```
SELECT * FROM dipendente d WHERE puntovendita = ?
```

**Operazione 11:**

```
SELECT *  
FROM puntoVendita pv, titolaresocio ts  
WHERE pv.codice = ts.puntovendita;
```

**Operazione 12:**

```
SELECT a.*, p.nome, cin.quantita  
FROM cliente c, effettuato_da eda, acquisto a, contenuto_in cin, prodotto p  
WHERE c.codicefiscale = ? AND c.codicefiscale = eda.cliente  
AND eda.acquisto = a.numeroordine AND a.numeroordine = cin.acquisto  
AND cin.prodotto = p.barcode  
GROUP BY a.numeroordine
```

**Operazione 13:**

```
SELECT p.barcode, p.nome, v.quantita + d.quantita AS Disponibilita Totale  
FROM prodotto p, vende v, depositato d  
WHERE p.barcode = v.prodotto AND p.barcode = d.prodotto  
GROUP BY p.barcode
```

**Operazione 14:**

```
SELECT p.barcode,p.nome"  
FROM prodotto p, vende v, puntovendita pv  
WHERE p.barcode = v.prodotto AND v.puntovendita = pv.codice AND v.quantita = 0 AND  
pv.codice = ?
```

**Operazione 15:**

```
SELECT p.barcode,p.nome  
FROM prodotto p, videogioco v  
WHERE p.barcode = v.prodotto AND YEAR(v.datarilascio) = ?
```

**Operazione 16:**

```
SELECT g.serie, COUNT(*) as numero_gadget  
FROM gadget g  
GROUP BY g.serie  
ORDER BY numero_gadget DESC
```

**Operazione 17:**

```
SELECT p.*  
FROM prodotto p, vende v, puntovendita pv  
WHERE p.barcode = v.prodotto AND v.puntovendita = pv.codice AND v.disponibilita > 0  
AND pv.codice = ?
```

**Operazione 18:**

```
SELECT *  
FROM prodotto p, console c  
WHERE p.barcode = c.prodotto AND c.famiglia LIKE '%?%'
```

**Operazione 19:**

```
SELECT a.*, p.*, v.*, c.*nome, cnt.tipo  
FROM acquisto a, contenuto_in cin, prodotto p, videogioco v, contenuto cnt, categoria c  
WHERE a.numeroordine = cin.acquisto AND cin.prodotto = p.barcode AND p.barcode =  
v.prodotto AND v.prodotto = c.videogioco AND v.prodotto = cnt.videogioco  
AND a.stato = 'Prenotato'
```

**Operazione 20:**

```
SELECT p.*, v.sconto  
FROM prodotto p, vende v, puntovendita pv  
WHERE p.barcode = v.prodotto AND v.puntovendita = pv.codice AND pv.codice = ?  
AND v.sconto > 0
```

**Operazione 21:**

```
SELECT * FROM acquisto a WHERE a.stato = 'In Consegna'
```

## Test dell'applicazione Java

Una volta avviata l'applicazione verrà mostrata a schermo un'interfaccia di tipo testuale che mostra tutte le operazioni disponibili

```
**** Benvenuto nel nostro database ****
**** Operazioni disponibili ****
[1] Inserisci un nuovo videogame
[2] Inserisci un nuovo gadget
[3] Inserisci una nuova console
[4] Inserisci un nuovo dipendente
[5] Inserisci un nuovo punto vendita
[6] Inserisci una nuova carta fedeltà
[7] Inserisci un cliente
[8] Elencare tutti i clienti in possesso di una carta fedeltà almeno di livello 3
[9] Elencare tutte le informazioni sui videogiochi la cui valutazione PEGI di età è 18
[10] Elencare tutti i dipendenti che lavorano presso un determinato punto vendita
[11] Creare un elenco di tutti i punti vendita con nome e cognome dei rispettivi titolari soci
[12] Elencare tutte le informazioni sugli acquisti effettuati da un determinato cliente, assieme al nome e alla quantità dei vari prodotti acquistati per ogni acquisto
[13] Elencare la disponibilità totale per ogni prodotto, che include sia la disponibilità nei punti vendita che quella disponibili in magazzini
[14] Creare un elenco dei prodotti la cui disponibilità presso un determinato punto vendita è 0
[15] Elencare tutti i videogiochi rilasciati in un determinato anno
[16] Creare un elenco che conteggia i gadget in base alla serie in ordine decrescente
[17] Elencare tutti i prodotti in vendita presso un determinato punto vendita
[18] Elencare tutte le console appartenenti alla stessa famiglia
[19] Elencare tutte le informazioni riguardanti i preordini, includendo tutte le informazioni sui videogiochi
[20] Elencare tutti i prodotti in sconto disponibili presso un punto vendita
[21] Elencare tutte le informazioni sugli ordini in consegna
[0] Esci dall'applicazione
Inserisci scelta:
```

Inserita la scelta, se necessario, verranno stampati a schermo i dati da dare in input alle query

```
Inserisci scelta: 10
|Inserisci codice negozio:
```

Infine verrà stampato il risultato della query

```
Inserisci codice negozio:
2
DBConnectionPool Connessione OK

employeNumber: 12345670 | nome: Franco | cognome: Franchino | data di nascita: 1977-02-05 | sesso: M | ore settimanali: 20
-----
employeNumber: 12345678 | nome: Franco | cognome: Franco | data di nascita: 1971-02-04 | sesso: M | ore settimanali: 40
-----
employeNumber: 12345679 | nome: Giorgia | cognome: Meloni | data di nascita: 1981-03-04 | sesso: F | ore settimanali: 40
```

Di seguito, altri esempi di query:

#### Operazione 11 (non richiede parametro)

```
codice: 1 | via: Via Michelangelo Buonarroti | civico: 15 | citta: Angri | provincia: SA | numero dipendenti: 3 | Eleonora | Buono  
codice: 2 | via: Corso Vittorio Emanuele II | civico: 65 | citta: Milano | provincia: MI | numero dipendenti: 5 | Paolo | Bonolis  
codice: 3 | via: Corso Imperatore | civico: 42 | citta: Roma | provincia: RO | numero dipendenti: 7 | Drako | Laura  
codice: 4 | via: Via Lemani Dalnaso | civico: 1 | citta: L'Aquila | provincia: AQ | numero dipendenti: 1 | Elena | Gervasoni  
codice: 5 | via: Via Mariano Tuccella | civico: 51 | citta: Bologna | provincia: BO | numero dipendenti: 3 | Bobbe | Malle
```

#### Operazione 17 (non richiede parametro)

```
prodotto: 1234567890123 | nome: The Legend of Zelda: Tears of the Kingdom | prezzo: 60,00  
prodotto: 1234567890126 | nome: Nintendo Switch OLED | prezzo: 350,00
```

#### Operazione 20

```
Inserisci scelta: 20  
Inserisci codice negozio:  
4  
DBConnectionPool Connessione OK  
prodotto: 1234567890123 | nome: The Legend of Zelda: Tears of the Kingdom | prezzo: 60,00 | sconto: 20
```