

# Relazione progetto Basi di Dati 2016/2017

[www.amazing.com](http://www.amazing.com)

**Paolo Eccher**

**Manfredi Smaniotto**

## **ABSTRACT**

Con la diffusione capillare di internet nelle case a partire dalla metà degli anni 90 si è osservata una enorme crescita dei commerci online. Sempre più produttori offrono questo servizio attorno al quale gira ormai una enorme quantità di denaro. Nel 2012 il commercio online ha generato 364,60 miliardi di USD solo negli Stati Uniti. Il record giornaliero di merce venduta da un solo sito è posseduto da Alibaba, gruppo cinese, che l'11 novembre 2014 ha generato vendite che ammontano a quasi 10 miliardi di dollari. È evidente che per gestire in maniera affidabile e efficiente questi negozi online, anche quelli il cui fatturato non arriva a miliardi di dollari, è indispensabile dotarsi di una base di dati. Ciò che descriviamo in questa relazione è una base di dati che gestisce un negozio online di medie/grandi dimensioni operante in 5 nazioni europee.

Ci si focalizzerà soprattutto sul lato logistico tenendo sotto controllo i magazzini e gli inventari contenenti gli oggetti in vendita. Inoltre il database manterrà al suo interno le informazioni riguardanti gli ordini, dando la possibilità di risalire ai dati dei clienti. Si implementeranno soluzioni anche per gestire il rapporto con i fornitori per quanto riguarda il rifornimento di prodotti.

## **ANALISI DEI REQUISITI**

La base di dati modella un negozio online che propone varie tipologie di prodotti. Si vogliono considerare le informazioni riguardanti gli ordini e gli oggetti nonché i dati degli utenti iscritti.

Per l'E-Shop vanno memorizzati l'indirizzo URL, l'indirizzo civico, il recapito dedicato all'assistenza clienti e per poter costruire i bilanci bisogna tener conto delle uscite, in particolare quelle per gli stipendi dei dipendenti e quelle riguardanti le spese di mantenimento delle strutture. L'E-Shop è responsabile dell'organizzazione degli ordini. Essi sono identificati da un ID e sarà memorizzato l'indirizzo e-mail dell'utente che lo ha effettuato, l'importo, la data e l'ora di esecuzione dell'ordine, l'URL del sito da cui esso è stato eseguito e l'indirizzo per la spedizione. Possono esservi ordini completati, in corso, in preparazione e non riusciti. La spedizione è svolta dai Corrieri, per questi serviranno nome, indirizzo e recapito telefonico. Ogni spedizione è inoltre mantenuta nel database memorizzandone l'ID dell'ordine, il nome del corriere e il numero di spedizione nell'eventualità che per un ordine avvengano più spedizioni di più pacchi.

Ogni ordine viene eseguito a seguito di un pagamento tramite carta di credito. Per le carte sono necessari informazioni riguardanti il circuito, il proprietario, la scadenza e il numero. Viene inoltre mantenuto nel database lo stato della carta per segnalare l'eventuale scadenza della stessa. La carta è necessaria per portare a termine una Transazione che è necessaria per far partire la spedizione di un ordine. Per ogni Transazione si tiene conto del mittente, il beneficiario della transazione e le varie componenti dell'importo, dunque dell'importo totale, il costo di spedizione, l'imposta sul valore aggiunto e l'eventuale sconto applicato. Vengono inoltre memorizzati l'ID dell'ordine, la mail dell'utente, il numero della carta utilizzata, il circuito e lo stato del pagamento.

Per gli utenti, identificati da un ID univoco si vogliono memorizzare le informazioni personali come nome, cognome, indirizzo e-mail. Dato che si intende proporre dei "premi" per gli utenti più attivi sull'E-Shop per ogni mese andiamo anche a memorizzare la somma del denaro speso nel mese corrente.

Ogni oggetto in vendita è individuato da un codice a barre e vanno memorizzate informazioni basilari come nome, prezzo, peso, colore, marca e tipologia (elettronica, abbigliamento...). Da tutti gli oggetti vengono selezionati quelli facenti parte di un ordine, a cui ci riferiremo come Selezione. Viene associato al codice di

ogni ordine l'id dell'oggetto ordinato e la sua quantità. Viene inoltre registrato il magazzino da cui viene prelevato l'oggetto.

L'inventario dell'e-shop è articolato in magazzini dislocati sul territorio. I magazzini sono identificati da un ID numerico e riportano la loro nazionalità e l'indirizzo in cui hanno sede. Per mantenere nel database la quantità di pezzi disponibili presenti in un magazzino sarà predisposta una tabella Istanza oggetto che conterrà, oltre al codice a barre del prodotto voluto, l'ID del magazzino in cui è notificata la disponibilità con la relativa quantità, le riserve minime di pezzi del prodotto da mantenere in magazzino e la nazione in cui si trova. Quando un oggetto viene esaurito va effettuata una richiesta di rifornimento ai Fornitori. Questi riforniscono solamente i prodotti di una marca e vanno memorizzate informazioni riguardo il recapito, la e-mail e l'ammontare delle spese mensili che è sono state fatte dall'e-shop presso il suddetto fornitore per facilitare il calcolo del bilancio.

## GLOSSARIO TERMINI

<b>TERMINE:</b>	E-Shop
<b>DESCRIZIONE:</b>	È il sito da cui vengono effettuati gli ordini. Gestisce gli ordini e l'organizzazione delle spedizioni. Deve inoltre mantenere aggiornato il proprio bilancio.
<b>SINONIMO:</b>	Negozi online
<b>COLLEGAMENTO:</b>	Ordine

<b>TERMINE:</b>	Selezione
<b>DESCRIZIONE:</b>	Sono gli oggetti ordinati con la relativa quantità per ogni ordine.
<b>SINONIMO:</b>	Assortimento
<b>COLLEGAMENTO:</b>	Ordine, Catalogazione

<b>TERMINE:</b>	Ordine
<b>DESCRIZIONE:</b>	È l'insieme degli oggetti, raggruppati per magazzino, con peso in caso di più oggetti complessivo non superiore 20kg. È ogni "pacco"
<b>SINONIMO:</b>	Ordinazione, Commissione
<b>COLLEGAMENTO:</b>	Utente, Corriere, Transazione, Selezione, E-Shop

<b>TERMINE:</b>	Oggetto
<b>DESCRIZIONE:</b>	È ogni cosa venduta sull'e-shop. Appartiene ad una categoria.
<b>SINONIMO:</b>	Merce, prodotto
<b>COLLEGAMENTO:</b>	Fornitore, Catalogazione

<b>TERMINE:</b>	Utente
<b>DESCRIZIONE:</b>	Account registrato sull'e-shop. Può effettuare ordini, scorrere i cataloghi online e gestire le proprie informazioni, tra cui i metodi di

	pagamento.
<b>SINONIMO:</b>	Cliente
<b>COLLEGAMENTO:</b>	Transazione, Ordine

<b>TERMINE:</b>	Catalogazione
<b>DESCRIZIONE:</b>	È l'insieme di tutti gli oggetti presenti nei magazzini
<b>SINONIMO:</b>	Inventario, Registro
<b>COLLEGAMENTO:</b>	Oggetto, Selezione

<b>TERMINE:</b>	Magazzino
<b>DESCRIZIONE:</b>	È il luogo da cui vengono spediti gli elementi di un ordine
<b>SINONIMO:</b>	Deposito
<b>COLLEGAMENTO:</b>	Catalogazione

<b>TERMINE:</b>	Fornitore
<b>DESCRIZIONE:</b>	Ente che si occupa di rifornire l'e-shop della merce.
<b>SINONIMO:</b>	Grossista
<b>COLLEGAMENTO:</b>	Oggetto

<b>TERMINE:</b>	Corriere
<b>DESCRIZIONE:</b>	È la società terza che si occupa della consegna della merce al cliente
<b>SINONIMO:</b>	Spedizioniere
<b>COLLEGAMENTO:</b>	Ordine

<b>TERMINE:</b>	Carta di credito
<b>DESCRIZIONE:</b>	Metodo di pagamento registrato dall'utente per compiere una transazione
<b>SINONIMO:</b>	bancomat, carta elettronica
<b>COLLEGAMENTO:</b>	Transazione

<b>TERMINE:</b>	Transazione
<b>DESCRIZIONE:</b>	Operazione di pagamento eseguita da un utente.
<b>SINONIMO:</b>	pagamento
<b>COLLEGAMENTO:</b>	Ordine, Utente, Carta

## PROGETTAZIONE CONCETTUALE

### FRASI RELATIVE AD "E-SHOP"

Per l'E-Shop vanno memorizzati l'indirizzo URL, l'indirizzo civico, il recapito dedicato all'assistenza clienti e per poter costruire i bilanci bisogna tener conto delle uscite, in particolare quelle per gli stipendi dei dipendenti e quelle riguardanti le spese di mantenimento delle strutture. L'E-Shop è responsabile dell'organizzazione degli ordini. Essi sono identificati da un ID e sarà memorizzato l'indirizzo e-mail dell'utente che lo ha effettuato, l'importo, la data e l'ora di esecuzione dell'ordine, l'URL del sito da cui esso è stato eseguito e l'indirizzo per la spedizione. Possono esservi ordini completati, in corso, in preparazione e non riusciti. La spedizione è svolta dai Corrieri, per questi serviranno nome, indirizzo e recapito telefonico. Ogni spedizione è inoltre mantenuta nel database memorizzandone l'ID dell'ordine, il nome del corriere e il numero di spedizione nell'eventualità che per un ordine avvengano più spedizioni di più pacchi.

#### E-SHOP:

- URL(KEY): string ( not null, UNIQUE )
- INDIRIZZO: string ( not null )
- RECAPITO\_ASSISTENZA: string ( not null )
- SOMMA\_STIPENDI\_DIPENDENTI: double
- COSTO\_GESTIONE\_MENSILE: double

#### ORDINE:

- ID(KEY): int ( not null, UNIQUE )
- DATAORA: date
- URL\_SITO ( FOREIGN KEY E\_SHOP.URL ): string
- EMAIL\_CLIENTE ( FOREIGN KEY Utente.Email ): string
- IMPORTO: double
- INDIRIZZO\_DI\_SPEDIZIONE: string ( not null )
- STATUS:string

#### CORRIERE:

- NOME(KEY):string (not null, UNIQUE)
- INDIRIZZO\_SEDE:string
- RECAPITO\_TELEFONICO:int

#### SPEDIZIONE:

- ID\_ORDINE (KEY, FOREIGN KEY Ordine.ID): string
- NOME\_CORRIERE (KEY, FOREIGN KEY Utente.Email): string
- NUMERO (KEY): int

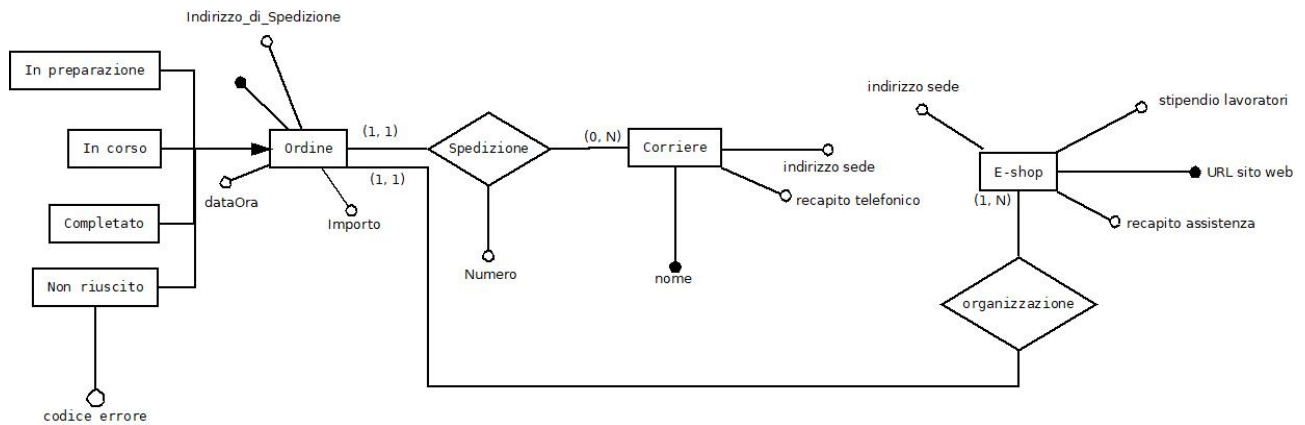
#### RELAZIONI:

##### E-SHOP-ORDINE: Organizzazione

- Un E-Shop gestisce da 1 ad N ordini
- Ogni ordine è gestito da un solo E-Shop nazionale

##### ORDINE-CORRIERI: Spedizione

- Un ordine è spedito da un solo corriere
- Un corriere può spedire nessuno o N ordini



### FRASI RELATIVE A “TRANSAZIONI”

Ogni ordine viene eseguito a seguito di un pagamento tramite carta di credito. Per le carte sono necessari informazioni riguardanti il circuito, il proprietario, la scadenza e il numero. Viene inoltre mantenuto nel database lo stato della carta per segnalare l'eventuale scadenza della stessa. La carta è necessaria per portare a termine una Transazione che è necessaria per far partire la spedizione di un ordine. Per ogni Transazione si tiene conto del mittente, il beneficiario della transazione e le varie componenti dell'importo, dunque dell'importo totale, il costo di spedizione, l'imposta sul valore aggiunto e l'eventuale sconto applicato. Vengono inoltre memorizzati l'ID dell'ordine, la mail dell'utente, il numero della carta utilizzata il circuito e lo stato del pagamento.

Per gli utenti, identificati da un ID univoco si vogliono memorizzare le informazioni personali come nome, cognome, indirizzo e-mail. Dato che si intende proporre dei “premi” per gli utenti più attivi sull'E-Shop per ogni mese andiamo anche a memorizzare la somma del denaro speso nel mese corrente.

#### TRANSAZIONE:

- ID\_TRANSAZIONE (KEY, FOREIGN KEY Ordine.ID) : int (not null, UNIQUE)
- MAIL\_UTENTE (FOREIGN KEY Utente.Email) : string
- ID\_ORDINE: int
- IVA: double
- IMPORTO\_NETTO: double
- BENEFICIARIO: string
- MITTENTE: string
- SCONTO: double
- NUMERO\_CARTA (FOREIGN KEY Carta\_di\_Credito.Numero) : int
- CIRCUITO\_CARTA (FOREIGN KEY Carta\_di\_Credito.Circuito) : string
- STATO: bool

#### UTENTE:

- EMAIL(KEY): string (not null, UNIQUE)
- NOME: char
- COGNOME: char
- SPESA MENSILE\_COMPLESSIVA: int

#### CARTA DI CREDITO:

- CIRCUITO: string (KEY, not null)
- NUMERO: string (KEY, not null)
- SCADENZA: Date (not null)
- STATO: bool
- PROPRIETARIO: string (not null)

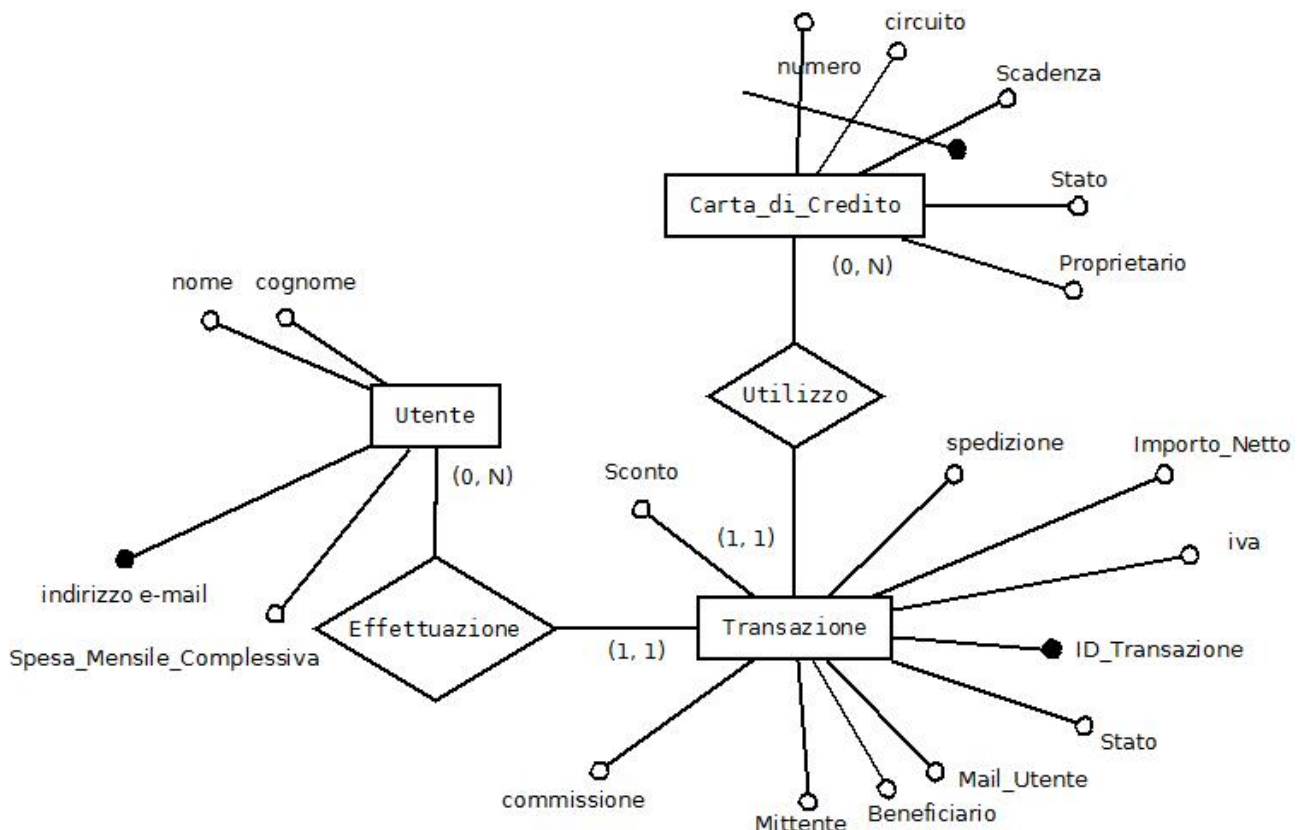
#### RELAZIONI:

##### CARTA-TRANSAZIONE: Utilizzo

- Una transazione necessita la presenza di una ed una sola carta.
- Una carta può essere utilizzata per un numero di transazioni da 0 a N.

##### UTENTE-ORDINE-TRANSAZIONE: Effettuazione

- Un ordine è effettuato da un solo utente.
- Un utente può effettuare un qualsiasi numero di ordini.
- Un utente svolge da 0 a N transazioni.
- Ogni transazione è valida per uno e un solo utente.
- Ad ogni ordine corrisponde una transazione.
- Per ogni transazione avvenuta avviene un ordine.



#### FRASI RELATIVE AD “OGGETTO”

Ogni oggetto in vendita è individuato da un codice a barre e vanno memorizzate informazioni basilari come nome, prezzo, peso, colore, marca e tipologia(elettronica, abbigliamento...). Da tutti gli oggetti vengono selezionati quelli facenti parte di un ordine, a cui ci riferiremo come Selezione. Viene associato al codice di ogni ordine l'id dell'oggetto ordinato e la sua quantità. Viene inoltre registrato il magazzino da cui viene prelevato l'oggetto.

L'inventario dell' e-shop è articolato in magazzini dislocati sul territorio. I magazzini sono identificati da un ID numerico e riportano la loro nazionalità e l'indirizzo in cui hanno sede. Per mantenere nel database la

quantità di pezzi disponibili presenti in un magazzino sarà predisposta una tabella Istanza oggetto che conterrà, oltre al codice a barre del prodotto voluto, l'ID del magazzino in cui è notificata la disponibilità con la relativa quantità, le riserve minime di pezzi del prodotto da mantenere in magazzino e la nazione in cui si trova. Quando un oggetto viene esaurito va effettuata una richiesta di rifornimento ai Fornitori. Questi riforniscono solamente i prodotti di una marca e vanno memorizzate informazioni riguardo il recapito, la e-mail e l'ammontare delle spese mensili che è sono state fatte dall'e-shop presso il suddetto fornitore per facilitare il calcolo del bilancio.

**OGGETTO:**

- CODICE\_A\_BARRE(KEY):string (not null, UNIQUE)
- NOME:string (not null)
- MARCA (FOREIGN KEY Fornitore.Nome) :string
- PREZZO:double
- PESO:double
- COLORE:string
- TIPOLOGIA:string

**ISTANZA OGGETTO:**(KEY È CODICE\_A\_BARRE, ID\_MAGAZZINO)

- NAZIONALITÀ (FOREIGN KEY Magazzino.Nazionalità) :string
- CODICE\_A\_BARRE (FOREIGN KEY Oggetto.Codice\_a\_Barre) :string
- ID\_MAGAZZINO (FOREIGN KEY Magazzino.ID) :int
- QUANTITÀ: int
- RISERVA:int

**MAGAZZINO:** (KEY È ID, NAZIONALITÀ)

- ID: Int
- NAZIONALITÀ: string
- INDIRIZZO: string

**FORNITORE:**

- NOME(KEY): string (not null, UNIQUE)
- EMAIL: string (UNIQUE)
- INDIRIZZO: string
- COMPLESSIVO\_RIFORNIMENTI: double

**SELEZIONE:**

- (KEY È ID\_OGGETTO, ID\_ORDINE, MAGAZZINO)
- ID\_Oggetto (FOREIGN KEY Catalogazione.Codice\_a\_Barre):string
- ID\_ORDINE (FOREIGN KEY Ordine.ID):int
- Quantità\_selezionata:int
- Magazzino (FOREIGN KEY Catalogazione.ID\_Magazzino): string

**RELAZIONI:**

**OGGETTO-FORNITORE:** Rifornimento

-Ogni oggetto è fornito da uno o più fornitori

-Un fornitore fornisce almeno un oggetto

**ISTANZA OGGETTO-OGGETTO:**Corrispondenza

- Un oggetto appartiene ad una e una sola catalogazione
- Alla catalogazione appartengono da 1 a N oggetti

#### ISTANZA OGGETTO-MAGAZZINO: Collocazione

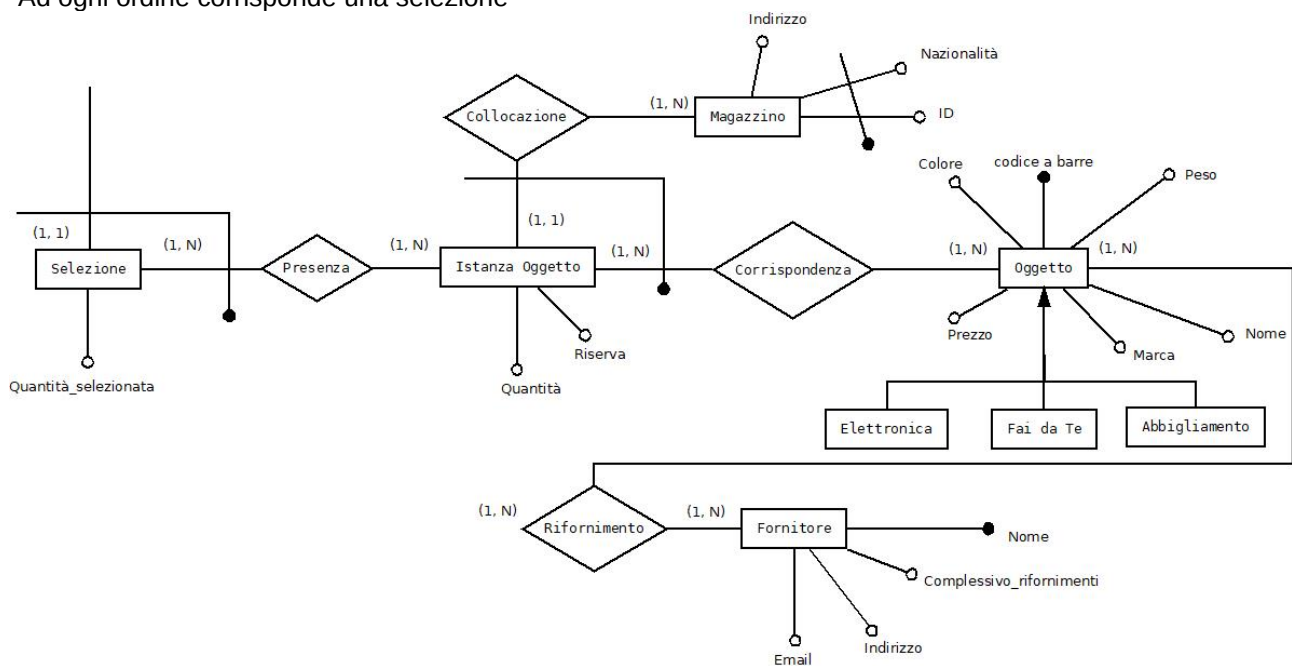
- Ad ogni istanza di un oggetto corrisponde un solo magazzino
- Ad un magazzino possono corrispondere da 1 a N istanze di oggetti

#### ISTANZA OGGETTO-SELEZIONE: Presenza

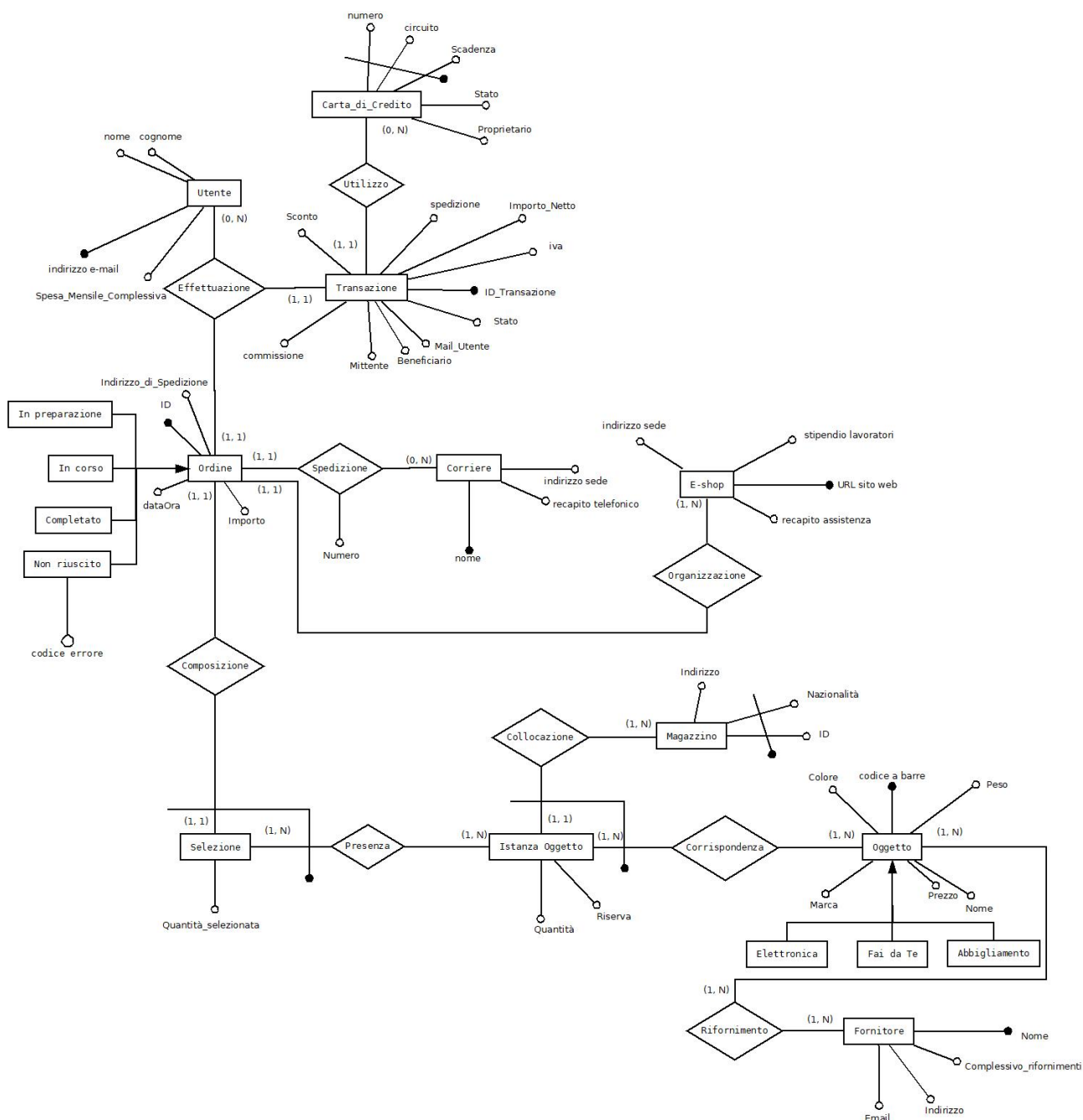
- In una selezione per un ordine vi sono da 1 a N istanze di oggetti
- Una istanza di un oggetto può essere selezionata in una o più selezioni per un ordine

#### SELEZIONE-ORDINE: Composizione

- Ad ogni selezione corrisponde un ordine
- Ad ogni ordine corrisponde una selezione







### VINCOLI NON ESPRIMIBILI:

- Ordine multiplo non può pesare più di 20kg.
- La spedizione degli ordini avviene solamente in seguito ad una transazione positiva.
- Spedizione gratuita per ordini maggiori di 29€.

### TAVOLA VOLUMI:

UTENTI	3'000'000
CARTE	3'000'000
OGGETTI	500'000
TRANSAZIONI	150'000
ORDINI	200'000

ELEMENTI SELEZIONATI	160'000
MAGAZZINI	15
INVENTARI NAZIONALI	5
FORNITORI	10'000
SPEDIZIONI	200'000
CORRIERI	30

## TAVOLA OPERAZIONI

**Operazione 1:** trovare le informazioni riguardanti tutti gli ordini ancora da spedire per ogni magazzino.

**Operazione 2:** trovare le informazioni e la quantità disponibile per ogni oggetto presente in ogni magazzino.

**Operazione 3:** trovare nome, marca, prezzo oggetti in vendita con la loro disponibilità per ogni nazione.

**Operazione 4:** ritornare il bilancio.

**Operazione 5:** numero ordini intestati ad un determinato indirizzo.

**Operazione 6:** trovare il numero di oggetti acquistati da una determinata marca.

**Operazione 7:** trovare l'utente che ha speso il più alto ammontare di denaro nel mese corrente.

**Operazione 8:** inserire un nuovo ordine.

OPERAZIONE	TIPO	FREQUENZA
Op. 1	B	2000 al giorno (ogni 5minuti per 15 magazzini)
Op. 2	B	10 al giorno
Op. 3	B	1 al secondo
Op. 4	B	20 al giorno
Op. 5	I	1 al mese
Op. 6	I	4 al mese
Op. 7	I	10 al giorno
Op. 8	I	1440 al giorno

## Progettazione Logica

### ANALISI RIDONDANZA

L'analisi della ridondanza sarà effettuata sull'attributo "Spesa\_Mensile\_Corrente" all'interno dell'entità Utente. Esso è coinvolto nelle operazioni 7 e 8.

**Operazione 7:** trovare l'utente che ha speso il più alto ammontare di denaro nel mese corrente. ( 10/giorno)

**Operazione 8:** inserire un nuovo ordine. ( 1440/giorno )

UTENTI	3'000'000
--------	-----------

<b>ORDINI</b>	<b>200'000</b>
---------------	----------------

Per memorizzare spesa mensile si usa un double che occupa 4byte. Pertanto, il costo di tenere per ogni utente la corrispondente spesa mensile è:

$$3000000 * 4B = 12000000B = \mathbf{12 MB}$$

**TAVOLA DEGLI ACCESSI CON RIDONDANZA:**

<b>OPERAZIONE 7</b>			
Concetto	Costrutto:	Accessi:	Tipo
UTENTI	E	3'000'000	L

L'operazione accede per 3 milioni di volte all' entità Utente. Gli accessi sono in lettura e l'operazione è effettuata 10 volte al giorno pertanto il costo giornaliero è:

$$C = 3'000'000 * 10 * 1 = \mathbf{30'000'000}$$

<b>OPERAZIONE 8</b>			
Concetto	Costrutto:	Accessi:	Tipo
UTENTI	E	1	S
EFFETTUATO	R	1	L
ORDINI	E	1	L

L'operazione accede a Ordini e Effettuato una volta in lettura e ad Utenti in scrittura. Essa viene effettuata 1440 volte al giorno pertanto il costo giornaliero è:

$$C = 1440 * ( 1 * 2 + 1 + 1 ) = \mathbf{5760}$$

**TAVOLA DEGLI ACCESSI SENZA RIDONDANZA:**

<b>OPERAZIONE 7</b>			
Concetto	Costrutto:	Accessi:	Tipo
UTENTI	E	3'000'000	L
EFFETTUATO	R	200'000	L
ORDINI	E	200'000	L

L'operazione accede a Utenti, Effettuato e Ordini in lettura un numero di volte pari al numero di righe delle suddette tabelle. Viene effettuata 10 volte al giorno pertanto il costo è pari a:

$$C = (3'000'000+200'000+200'00)*1 *10 = \mathbf{34000000}$$

<b>OPERAZIONE 8</b>			
Concetto	Costrutto:	Accessi:	Tipo
ORDINI	E	1	S

L'operazione accede a Ordini una volta in scrittura e viene effettuata 1440 volte al giorno. Il costo è dunque:

$$C = 1440*(1*2) = \mathbf{2880}$$

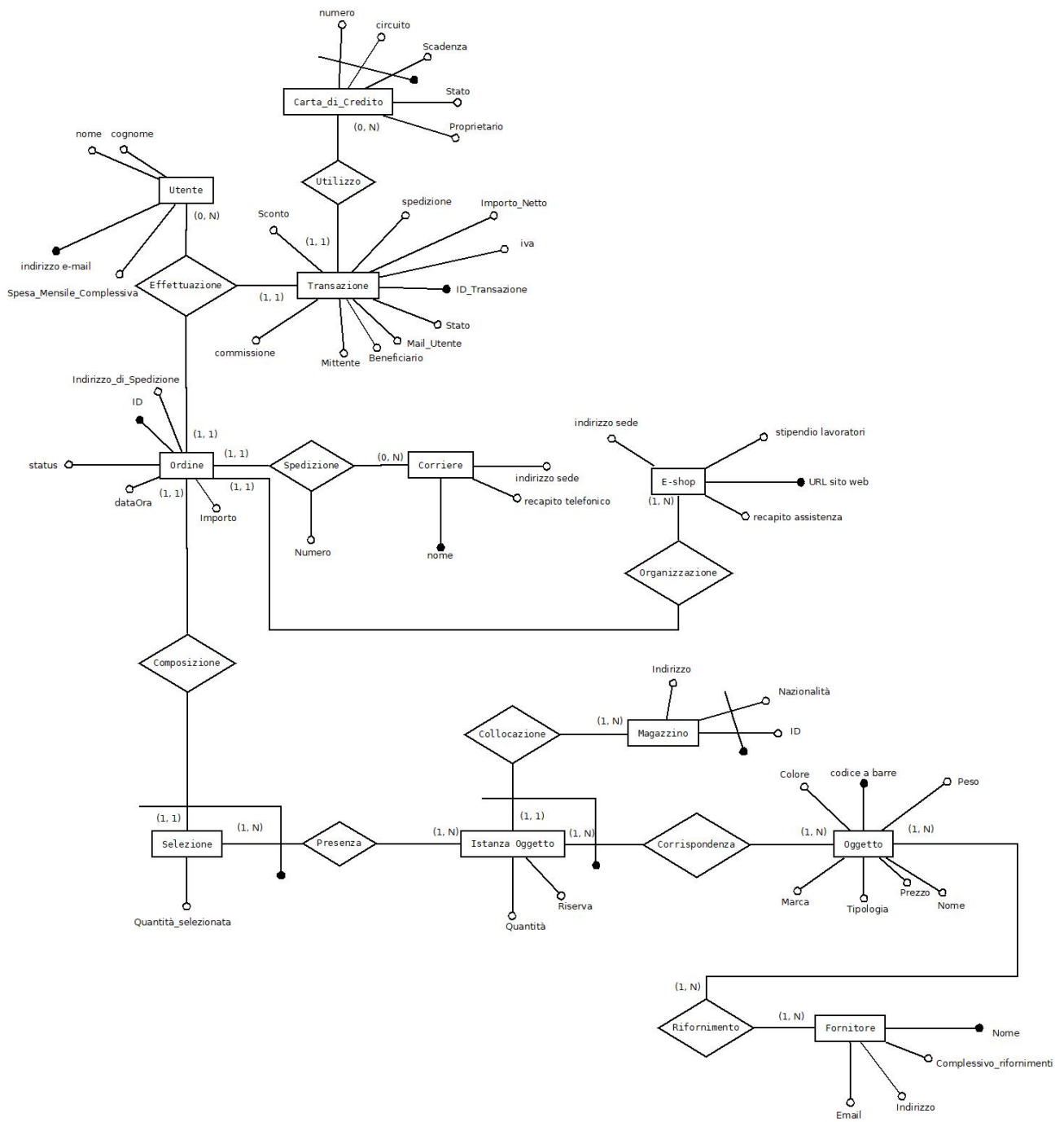
### **Conclusioni ridondanza**

A causa di uno sbilanciamento verso il numero di utenti (che sono 15 volte più del numero di ordini) si ha che il peso dell'operazione 7 è maggiore di quello dell'operazione 7. Di conseguenza è più conveniente tenere la ridondanza. Infatti con un costo di soli 12MB si risparmiano 4 milioni di accessi giornalieri nei i concetti Effettuato ed Ordine. Il mantenimento della coerenza del dato non risulta essere particolarmente esoso.

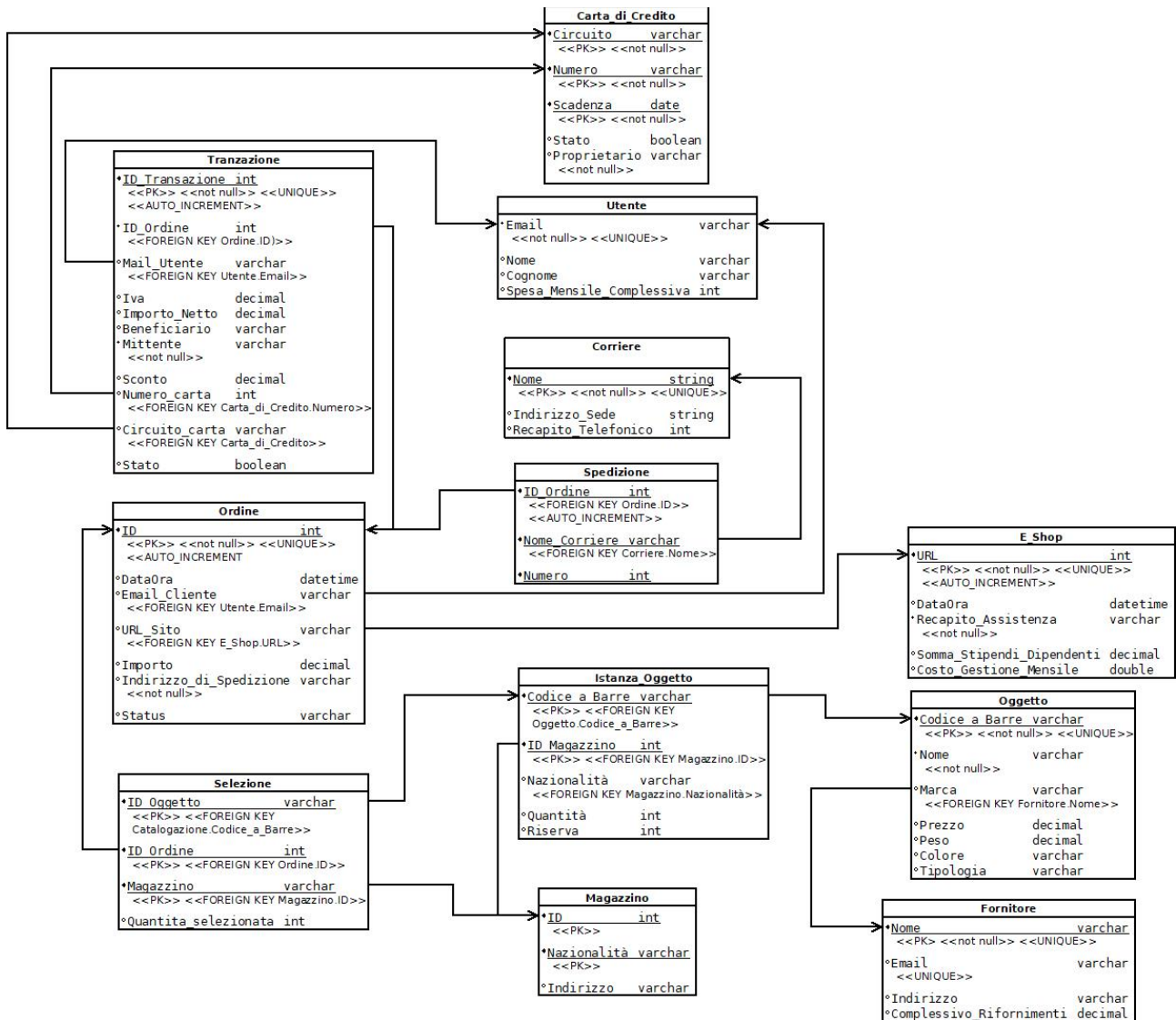
### **ELIMINAZIONE GENERALIZZAZIONI**

Si analizzano ora le due generalizzazioni presenti nello schema ER non ristrutturato.

- **GENERALIZZAZIONE ORDINE:** Si è deciso di accorpare le entità figlio verso l' entità genitore in quanto le operazioni non non fanno molta distinzione tra le occorrenze e gli attributi di "In preparazione", "In corso", "Completato" e "Non riuscito".
- **GENERALIZZAZIONE OGGETTO:** Si è deciso di procedere come nel caso precedente. Si sono dunque accorpati



## ■ Schema ER ristrutturato



■ schema logico

## PARTE SQL

#QUERY 1

#Seleziona tutti gli attributi di tutti gli oggetti disponibili in  
#ogni magazzino(sono 17)

```

select *
from Oggetto
where Codice_a_Barre IN (
    select Codice_a_Barre
    from Istanza_Oggetto
    GROUP BY Codice_a_Barre
    HAVING COUNT(ID_Magazzino)=17
)

```

;

```

+-----+-----+-----+-----+-----+-----+-----+
| Codice_a_Barre | Nome      | Marca | Prezzo | Peso | Colore | Tipologia |
+-----+-----+-----+-----+-----+-----+-----+
| 1              | iPhone 7 | Apple | 729.00 | 0.23 | Nero   | Smartphone |
+-----+-----+-----+-----+-----+-----+-----+

```

## #QUERY 2

#Selezionare nome, marca, prezzo degli oggetti in vendita con la  
#loro disponibilità per i magazzini in Italia

```
select o.Nome, o.Marca, o.Prezzo, sum(c.Quantita) AS Disponibilita_Italia
from Oggetto o join Istanza_Oggetto c ON o.Codice_a_Barre=c.Codice_a_Barre
where c.Nazionalita='Italia'
GROUP BY o.Codice_a_Barre
;
```

Nome	Marca	Prezzo	Disponibilita_Italia
IPhone 7	Apple	729.00	378
All Star	Converse	60.00	217
All Star	Converse	60.00	196
Cover iPhone 7	Cellular Line	9.99	133
Cover Galaxy S7	Cellular Line	9.99	219
IPhone 7	Apple	734.00	415
Galaxy S7	Samsung	739.00	124
Galaxy S7 Edge	Samsung	779.99	148
ZenBook	ASUS	1899.00	237
Originals	Adidas	99.00	211
Stan Smith	Adidas	59.00	187

## #QUERY 3

#Trovare tutti gli utenti che hanno fatto 2 ordini da almeno 30  
#euro spediti a indirizzi diversi e con corrieri diversi

```
select distinct u.Nome AS Nome, u.Cognome AS Cognome
from Utente u, Ordine o
where u.Email=o.Email_Cliente AND o.Email_Cliente IN
(
select distinct o1.Email_Cliente
from Ordine o1, Ordine o2, Spedizione s1, Spedizione s2
where o1.Email_Cliente=o2.Email_Cliente
AND o1.ID<>o2.ID
AND o1.Indirizzo_di_Spedizione<>o2.Indirizzo_di_Spedizione
AND o1.Importo>=30 AND o2.Importo>=30
AND s1.ID_Ordine=o1.ID AND s2.ID_Ordine=o2.ID
AND s1.Nome_Corriere=s2.Nome_Corriere
)
;
```

Nome	Cognome
Mario	Rossi

## #QUERY 4

#Ritorna il numero di ordini intestati ad un determinato indirizzo

```

select COUNT(*) as Numero_Ordini
from Ordine
where Indirizzo_di_Spedizione = 'Piazza Garibaldi 54, Padova'
GROUP BY Indirizzo_di_Spedizione
;
+-----+
| Numero_Ordini |
+-----+
|          5 |
+-----+

```

#### #QUERY 5

*#Trovare il numero di oggetti acquistati da una determinata marca*

```

select sum(s.Quantita_selezionata)+sum(c.Quantita) AS Totale_Oggetti_Acquistati
from Istanza_Oggetto c, Selezione s, Oggetto o
where o.Codice_a_Barre=s.ID_Oggetto
AND o.Codice_a_Barre=c.Codice_a_Barre AND o.Marca='Samsung'
GROUP BY o.Marca
;
+-----+
| Totale_Oggetti_Acquistati |
+-----+
|          1851 |
+-----+

```

#### #QUERY 6

*#Trovare l'utente che ha speso il più alto ammontare di denaro nel mese corrente*

```

select *
from Utente
where Spesa_Mensile_Complessiva=
(
    select max(u.Spesa_Mensile_Complessiva)
    from Utente u
)
;
+-----+-----+-----+-----+
| Email | Nome | Cognome | Spesa_Mensile_Complessiva |
+-----+-----+-----+-----+
| pedro.sanchez@gmail.com | Pedro | Sanchez | 5235 |
+-----+-----+-----+-----+

```

#### #QUERY 7

*#Trovare tutte le carte usate da un utente*

```

select distinct c.Circuito AS Circuito, c.Numero AS Numero, c.Scadenza AS
Scadenza, c.Stato AS Stato, c.Proprietario AS Proprietario
from Carta_di_Credito c, Transazione t
where c.Numero=t.Numero_carta AND c.Circuito=t.Circuito_carta AND t.ID_Ordine IN
(
    select o.ID

```



```

        from Ordine o, Utente u
        where o.Email_Cliente=u.Email
        AND u.Email="mario.rossi@gmail.com"
    )

```

```

;
+-----+-----+-----+-----+-----+
| Circuito | Numero          | Scadenza   | Stato | Proprietario |
+-----+-----+-----+-----+-----+
| MasterCard | 5208928454728826 | 2020-05-01 | 1     | Mario Rossi  |
+-----+-----+-----+-----+-----+

```

#### #Trigger 1

*#Trigger che in seguito ad un ordine aggiorna la spesa mensile corrente  
#dell'utente che ha effettuato il tal ordine. Se l'importo è inferiore a 20 si  
#attribuisce solamente l'importo effettivo, se invece l'importo dell'ordine è  
#compreso tra 20 e 50 oltre all'importo effettivo si aggiungono 10euro bonus.  
#Se l'ordine effettuato è >50 euro il bonus vale 25.*

```

CREATE TRIGGER AggiornaSpesa AFTER INSERT ON Ordine
FOR EACH ROW
BEGIN
UPDATE Utente
IF ( new.Importo < 20 )
SET Spesa_Mensile_Complessiva = Spesa_Mensile_Complessiva + new.Importo
ELSE IF ( new.Importo <50 )
SET Spesa_Mensile_Complessiva = Spesa_Mensile_Complessiva + new.Importo+10
ELSE
SET Spesa_Mensile_Complessiva = Spesa_Mensile_Complessiva + new.Importo+25
WHERE Email = new.Email_Cliente;
END

```

#### #Trigger 2

*#Trigger che prima dell'inserimento di una riga nella tabella Ordine verifica  
#che il peso complessivo di tutti i prodotti presenti nell'ordine non superi la  
#soglia dei 20 kg.  
#Nel caso in cui tale soglia non venga superata non viene applicata alcuna  
#modifica.  
#Altrimenti viene lanciato un messaggio di errore e viene bloccato  
#l'inserimento della riga nella tabella.*

```

CREATE TRIGGER CheckPeso
BEFORE INSERT ON Ordine
FOR EACH ROW
BEGIN
DECLARE Weight INT;
SELECT Sum(o.Peso * s.Quantita_selezionata) INTO Weight
FROM Selezione s JOIN Oggetto o ON (s.ID_Ordine=NEW.ID)
WHERE s.ID_Oggetto=o.Codice_a_Barre;
IF (Weight>20) THEN
    SIGNAL SQLSTATE 'ERROR'
SET
    MESSAGE_TEXT = `Weight of the package is too high`,
    MYSQL_ERRNO = `45000`;
END IF;
END

```

### #Trigger 3

*#Trigger che aggiunge le spese di spedizione (che ammontano a 5 euro)  
#agli ordini che ammontano a meno di 29euro*

```
CREATE TRIGGER Spese_di_Spedizione
BEFORE INSERT ON Transazione
FOR EACH ROW
BEGIN

DECLARE Spesa INT ;
SELECT o.Importo INTO Spesa
FROM Ordine o
WHERE new.ID_Ordine=o.ID ;

IF (Spesa<29.00) THEN
    SET new.Importo_Netto=new.Importo_Netto+5 ;
END IF ;

END
```

### #FUNCTION 1

*#Ritorna l'utente che ha effettuato il maggior numero di ordini contenenti un  
#oggetto di una certa marca, che è passata alla funzione.*

```
CREATE FUNCTION UtenteMaxNumeroProdottiMarca( Marca varchar(191) )
RETURNS VARCHAR(30)
BEGIN
DECLARE emailutente VARCHAR(30);
SELECT u.Email INTO emailutente
FROM Utente u JOIN Ordine o JOIN Selezione s JOIN Oggetto og
WHERE u.Email=o.Email_Cliente AND o.ID=s.ID_Ordine AND
s.ID_Oggetto=og.Codice_a_Barre AND og.Marca=Marca
GROUP BY u.Email
ORDER BY COUNT(*) DESC
LIMIT 1;
RETURN emailutente;
END
```

```
+-----+
| UtenteMaxNumeroProdottiMarca("Apple") |
+-----+
| filippo.neri@yahoo.com                |
+-----+
```

### #FUNCTION 2

*#La funzione restituisce la spesa effettuata in totale dall'utente passatogli*

```
CREATE FUNCTION CalcolaSpesaUtente( Email VARCHAR(191) )
RETURNS DECIMAL ( 10, 2)
BEGIN
DECLARE spesa DECIMAL ( 10, 2);
SELECT SUM(o.Importo) INTO spesa
FROM Ordine AS O JOIN Utente AS U ON O.Email_Cliente=U.Email
WHERE U.Email=Email;
RETURN spesa;
END
```

```
+-----+
| CalcolaSpesaUtente("mario.rossi@gmail.com") |
+-----+
|                                     3409.79 |
+-----+
```

### #FUNCTION 3:

*#ritorna il bilancio dell'anno in corso, sottraendo agli incassi le spese di gestione*

```
CREATE FUNCTION CalcolaBilancio()
RETURNS DECIMAL(20,2)
BEGIN
DECLARE stipendi DECIMAL(20,2) ;
DECLARE speseGestione DECIMAL(20,2) ;
DECLARE ordini DECIMAL(20,2) ;
DECLARE mese INT(2);
DECLARE anno INT(4);
DECLARE risultato DECIMAL(21,2);
SELECT MONTH(NOW()) INTO mese ;
SELECT YEAR(NOW()) INTO anno ;
SELECT SUM(Somma_Stipendi_Dipendenti)*mese AS STIP INTO stipendi
FROM E_Shop ;
SELECT SUM(Costo_Gestione_Mensile)*mese AS GESTIONE INTO speseGestione
FROM E_Shop;
SELECT SUM(Importo) INTO ordini
FROM Ordine
WHERE MONTH(DataOra)<=mese && YEAR(DataOra)=anno ;
SELECT ordini-stipendi-spesesGestione AS RET INTO risultato;
RETURN risultato ;
END
$$
+-----+
| CalcolaBilancio() |
+-----+
|          -2048468.49 |
+-----+
```

### #PROCEDURE 1

*#La procedura azzera le spese mensili complessive di tutti gli utenti dell'E-shop. essa viene richiamata una volta ogni mese da un event appositamente creato (non è stato possibile testarlo poiché richiede i permessi di root per essere inserito nello scheduler)*

```
DROP EVENT IF EXISTS SCHEDULER_CONCORSO;
CREATE EVENT SCHEDULER_CONCORSO
ON SCHEDULE EVERY 1 MONTH
DO
CALL Azzera_Spese ;

DROP PROCEDURE IF EXISTS Azzera_Spese;
DELIMITER $$
CREATE PROCEDURE Azzera_Spese()
BEGIN
UPDATE Utente SET Spesa_Mensile_Complessiva = 0;
END
```