

Università degli Studi di Trieste
Intelligenza Artificiale e Data Analytics

Progetto di Basi di Dati
"Sito web Nike.com Germania"

SM3201132

Piero Cancian

31 maggio 2022

Presentazione progetto

Si vuole realizzare un data base per il sito online della multinazionale statunitense Nike, produttrice di calzature, abbigliamento e accessori sportivi, considerando solamente il suo operato in Germania. Il sito deve permettere la vista degli articoli all'utente che si collega e in caso di acquisto operare il trasporto della merce dal magazzino al punto di ritiro, che coincide con la residenza del cliente.

Dei magazzini si conosce codice, via, città e numero di telefono.

I prodotti hanno un codice, un nome, una tipologia (maglia, pantalone, scarpa, ecc.), una taglia, un prezzo di acquisto, un'immagine con opportuna descrizione e una data di produzione (indicata dall'anno). Possono essere firmati Nike oppure far parte di una collaborazione con altri brand di moda.

Per accedere al sito e per effettuare acquisti ogni utente deve registrarsi con username, password, e-mail, numero di telefono e indirizzo di residenza (via e città), dove farà recapitare il proprio ordine. In più si vuole sapere nome, cognome, data di nascita e codice fiscale.

Per ogni ordine, caratterizzato dal numero, si tiene traccia della data di acquisto, dei prodotti acquistati, dello stato (confermato, annullato, pagato, spedito, consegnato), dei dati dell'utente, del metodo di pagamento e di spedizione.

L'operazione di pagamento, caratterizzata da un codice, può essere effettuata in contanti, al momento della consegna, o carta di credito, quando si effettua l'ordine, e si vuole tenere traccia della data del pagamento e dell'importo dell'ordine.

La spedizione, caratterizzata da un codice, può essere affidata a un ente esterno, oppure a un corriere Nike, e si vuole tenere traccia della data di spedizione e dati dell'articolo consegnato.

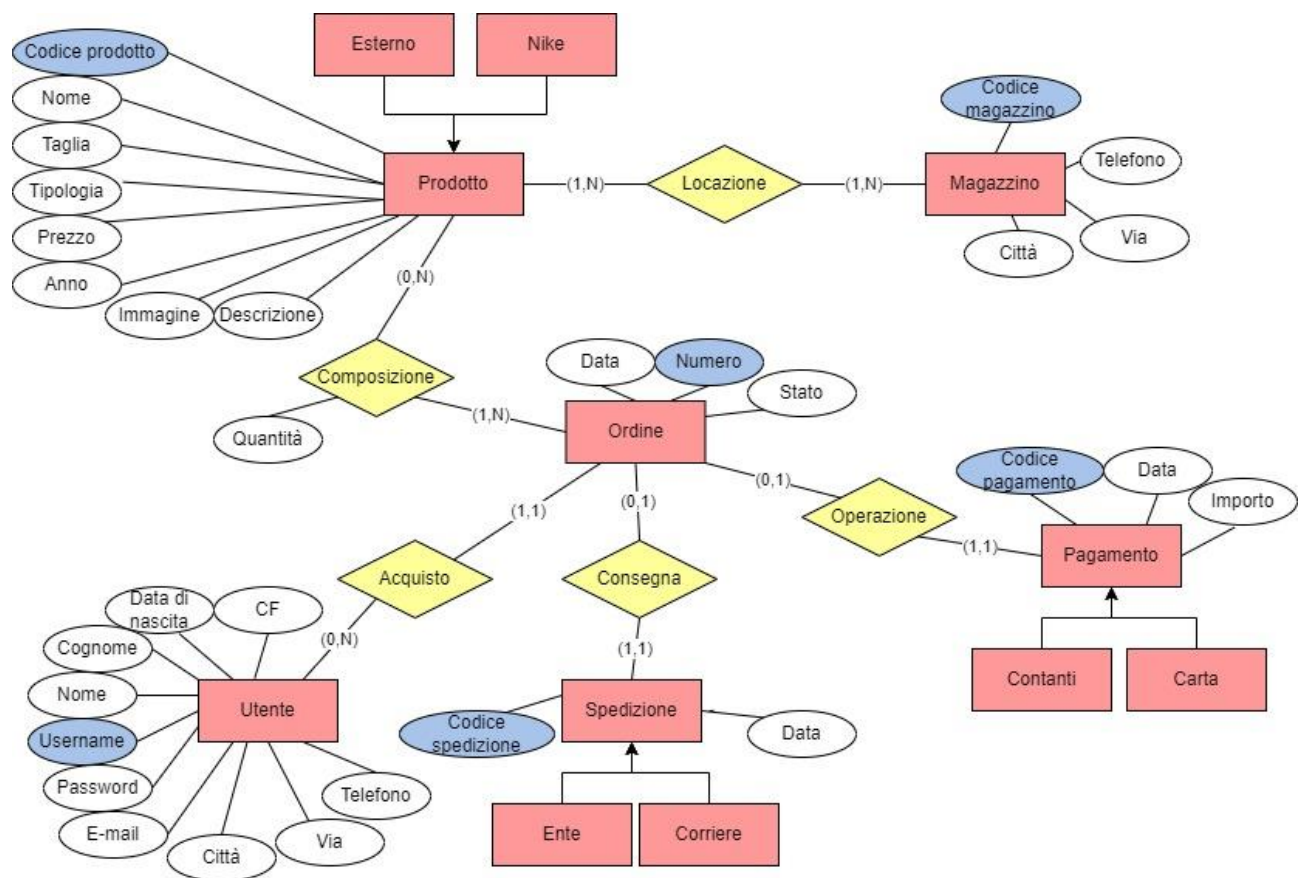
Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamento
Prodotto	Oggetto in vendita	Merce, articolo	Ordine
Utente	Persona che accede al sito	Cliente	Ordine
Magazzino	Luogo di produzione degli articoli		Prodotto
Ordine	Ordine		Prodotto, Pagamento, Utente, Spedizione
Pagamento	Pagamento dell'ordine		Ordine
Spedizione	Spedizione del prodotto al cliente		Ordine

Operazioni di interesse

- Si vogliono ottenere i dati degli ordini annullati e dei relativi acquirenti
(1 volta al mese)
- Si vogliono ottenere i dati degli utenti senza acquisti
(1 volta alla settimana)
- Ricerca degli ordini che coinvolgono una determinata tipologia di prodotto avvenute nell'ultima settimana, ordinate per dimensione dell'importo decrescente
(1 volta a settimana)
- Si vogliono ottenere le prime 3 tipologie di prodotto più vendute nell'ultimo mese
(2 volta a settimana)
- Si vogliono ottenere i dati degli utenti con almeno 5 ordini effettuati nell'ultimo anno lavorativo
(1 volta l'anno)

Schema Entity-Relationship



Dizionario dei Dati (Entità)

Entità	Descrizione	Attributi	Identificatore
Magazzino	Luogo di produzione merce	CodiceMagazzino, Via, Città, Numero di telefono	CodiceMagazzino
Prodotto	Articolo in vendita	CodiceProdotto, Nome, Taglia, Tipologia, Prezzo, Anno, Immagine, Descrizione	CodiceProdotto
Esterno	Prodotto realizzato da una collaborazione	CodiceProdotto, Nome, Taglia, Tipologia, Prezzo, Anno, Immagine, Descrizione	CodiceProdotto
Nike	Prodotto realizzato solo da Nike	Codice Prodotto, Nome, Taglia, Tipologia, Prezzo, Anno, Immagine, Descrizione	CodiceProdotto
Ordine	Ordine	Numero, Data, Stato	Numero
Utente	Visitatore del sito web	Nome, Cognome, Data di nascita, Codice fiscale, Via, Città, Username, Password, E-mail, Telefono	Username
Pagamento	Operazione di pagamento	CodicePagamento, Data, Importo	CodicePagamento
Contanti	Metodo di pagamento	CodicePagamento, Data, Importo	CodicePagamento
Carta di credito	Metodo di pagamento	CodicePagamento, Data, Importo	CodicePagamento
Spedizione	Operazione di trasferimento merce	CodiceSpedizione, Data	CodiceSpedizione
Ente	Corriere esterno a Nike	CodiceSpedizione, Data	CodiceSpedizione
Corriere	Corriere interno di Nike	CodiceSpedizione, Data	CodiceSpedizione

Dizionario dei Dati (Relationships)

Relazione	Descrizione	Composizione	Attributi
Operazione	Pagamento dell'ordine	Ordine, Pagamento	
Locazione	Dove si trova il prodotto	Prodotto, Magazzino	
Acquisto	Acquisto del prodotto	Ordine, Utente	
Consegna	Consegna dell'ordine	Ordine, Spedizione	
Composizione	Composizione dell'ordine	Prodotto, Ordine	Quantità

Vincoli Non Esprimibili

- L'username deve essere di almeno 10 caratteri e la password di almeno 8 caratteri
- Gli articoli la cui data di produzione risale a più di 2 anni fa non vengono più messi in vendita

Tabella dei Volumi

Concetto	Tipo	Volume
Magazzino	E	35
Prodotto	E	1000
Esterno	E	100
Nike	E	900
Ordine	E	200000
Cliente	E	60000
Spedizione	E	198000
Ente	E	38000
Corriere	E	160000
Pagamento	E	195000
Contanti	E	5000
Carta	E	190000
Locazione	R	2500
Operazione	R	195000
Acquisto	R	200000
Consegna	R	198000
Composizione	R	450000

Analisi ridondanze

E' presente una ridondanza causata dall'attributo 'importo' dell'entità 'pagamento', esso è un attributo derivabile dall'attributo 'prezzo' dell'entità 'prodotto'.

In seguito all'analisi delle operazioni che si vogliono eseguire sul database, in particolare per il fatto che c'è un basso numero di operazioni che richiedono l'importo di un ordine effettuato, si giunge alla conclusione che mantenere la ridondanza sia un appesantimento non necessario per il sistema. Si sceglie quindi di eliminare l'attributo 'importo'.

Eliminazione generalizzazioni

Sono presenti tre generalizzazioni, totali ed esclusive, riguardanti le seguenti entità: prodotto, pagamento, spedizione.

Si decide di accorpate i figli nel genitore, in tutti e tre i casi, poiché gli accessi sono contestuali ed inoltre nessuna delle entità figlie presenta attributi diversi o relazioni separate.

'Esterno' e 'Nike' saranno rappresentati dall'attributo 'marchio' in 'prodotto'.

'Ente' e 'Corriere' saranno rappresentati dall'attributo 'mezzo' in 'spedizione'.

'Contanti' e 'Carta' saranno rappresentati dall'attributo 'metodo' in 'pagamento'.

Partizionamento entità

Si decide di partizionare l'entità 'utente' in due nuove entità chiamate 'utente', che avrà come chiave primaria 'username', e 'dati anagrafici', che avrà come chiave primaria 'codice fiscale'. In questo modo si vogliono separare gli accessi riguardanti i dati del profilo utente, utilizzati per l'accesso e attraverso i quali l'azienda promuove i propri prodotti, dai dati personali, che vengono inseriti solamente nella fase finale dell'ordine.

Scelta identificatori primari

Magazzino: si usa l'attributo 'codice magazzino'

Prodotto: si usa l'attributo 'codice prodotto'

Utente: si usa l'attributo 'username'

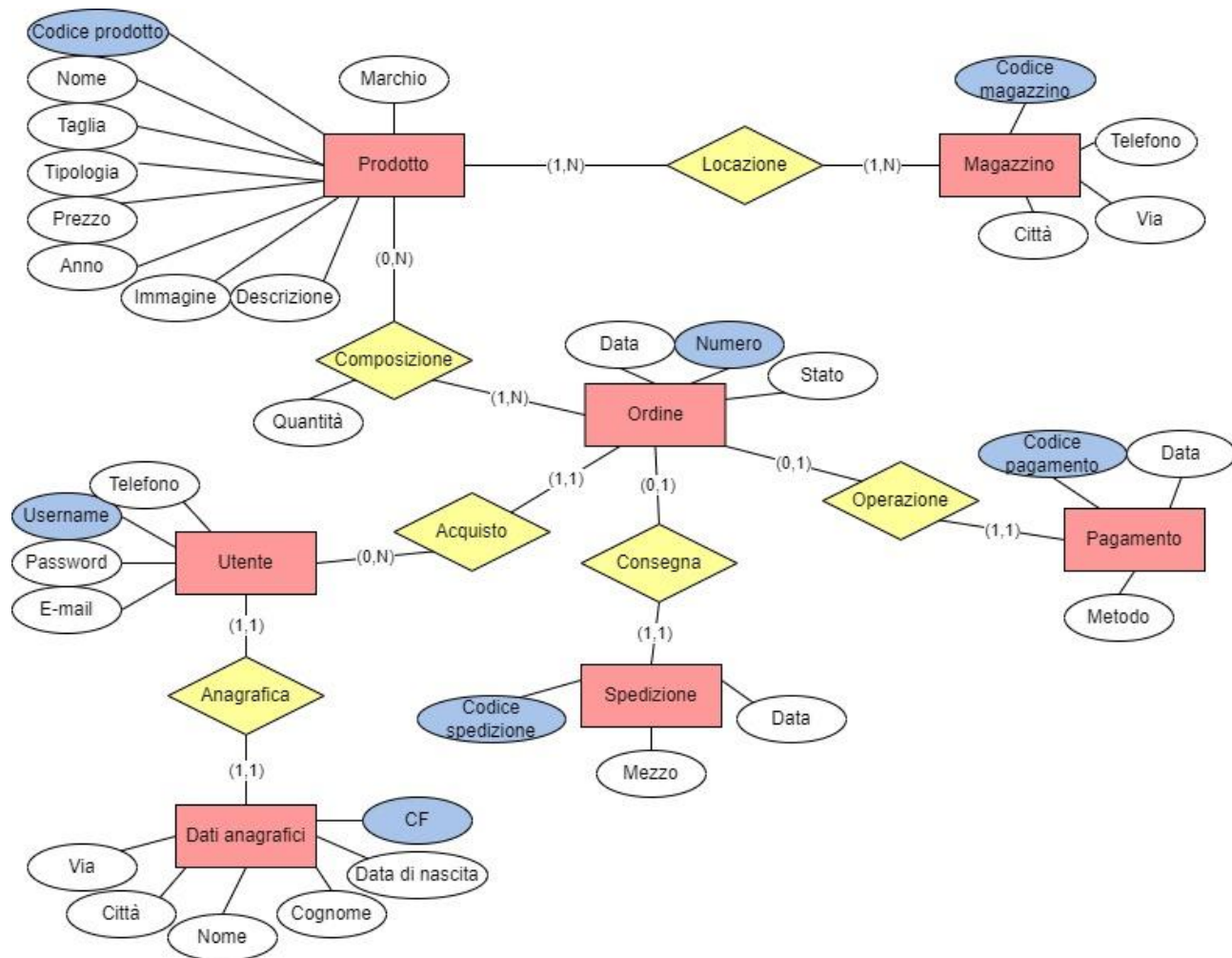
Ordine: si usa l'attributo 'numero'

Pagamento: si usa l'attributo 'codice pagamento'

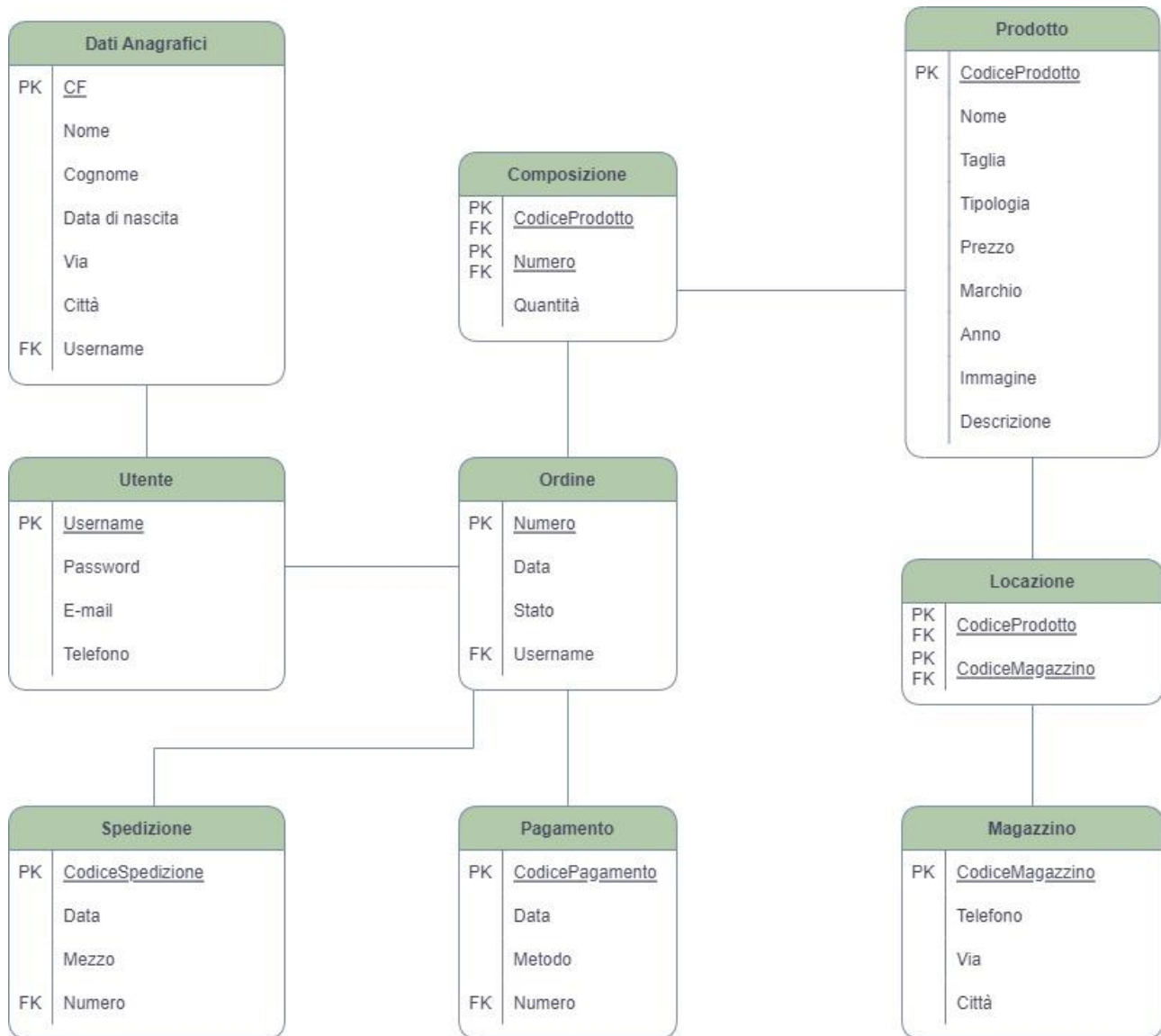
Spedizione: si usa l'attributo 'codice spedizione'

Dati anagrafici: si usa l'attributo 'codice fiscale'

Schema Entity-Relationship Ristrutturato



Schema Logico



Magazzino (Codice Magazzino, Telefono, Via, Città)

Locazione (Codice Prodotto, Codice Magazzino)

Prodotto (Codice Prodotto, Nome, Taglia, Tipologia, Prezzo, Marchio, Anno, Immagine, Descrizione)

Composizione (Codice Prodotto, Numero, Quantità)

Ordine (Numero, Data, Stato, Username)

Utente (Username, Password, E-mail, Telefono)

Dati anagrafici (CF, Nome, Cognome, Data di nascita, Via, Città, Username)

Spedizione (Codice Spedizione, Data, Mezzo, Numero)

Pagamento (Codice Pagamento, Data, Metodo, Numero)

Normalizzazione

Il database è in prima forma normale: tutti gli attributi sono atomici.

Il database è in seconda forma normale: ciascun attributo dipende strettamente dalla chiave primaria.

Il database è in terza forma normale: ogni attributo dipende unicamente dalla chiave primaria.

Query SQL

Si vogliono ottenere i dati degli ordini annullati e dei relativi acquirenti

```
DELIMITER $$
CREATE PROCEDURE sp_OrdiniAnnullati()
BEGIN
    SELECT numero, stato, data, da.username, nome, cognome, via, città FROM
    ordine o
    INNER JOIN utente u ON o.username = u.username
    INNER JOIN datiAnagrafici da ON u.username = da.username
    WHERE o.stato = 'annullato';
END$$
DELIMITER ;
```

Si vogliono ottenere i dati degli utenti senza acquisti

```
DELIMITER $$
CREATE PROCEDURE sp_NoAcquisti()
BEGIN
    SELECT u.username, u.email, u.telefono FROM utente u
    WHERE u.username NOT IN (SELECT o.username FROM ordine o);
END$$
DELIMITER ;
```

Ricerca degli ordini che coinvolgono una determinata tipologia di prodotto avvenute nell'ultima settimana, ordinate per dimensione dell'importo decrescente

```
DELIMITER $$
CREATE PROCEDURE sp_OrdiniTipologia(IN tipo VARCHAR(45))
BEGIN
    SELECT p.codiceProdotto, p.nome, p.tipologia, o.username, o.numero, o.data,
    o.stato, sum(c.quantità*p.prezzo) as tot FROM ordine o
    INNER JOIN composizione c USING (numero)
    INNER JOIN prodotto p USING (codiceProdotto)
    WHERE p.tipologia = tipo and o.data > CURDATE()-7
    GROUP BY p.codiceProdotto
    ORDER BY tot DESC;
END $$
DELIMITER ;
```

Si vogliono ottenere le prime 3 tipologie di prodotto più vendute nell'ultimo mese

```
DELIMITER $$
CREATE PROCEDURE sp_PodioTipologia()
BEGIN
    SELECT tipologia, sum(quantità) as tot FROM ordine
    INNER JOIN composizione c USING (numero)
    INNER JOIN prodotto p ON c.codiceProdotto = p.codiceProdotto
    WHERE ordine.data > CURDATE() - 31
    GROUP BY tipologia
    ORDER BY tot DESC
    LIMIT 3;
END$$
DELIMITER ;
```

Si vogliono ottenere i dati degli utenti con almeno 5 ordini effettuati nell'ultimo anno lavorativo

```
DELIMITER $$
CREATE PROCEDURE sp_ClientiCinquePiuOrdini()
BEGIN
    SELECT username, da.nome, da.cognome, da.via, da.città, count(*) as
    numOrdine FROM ordine
    INNER JOIN utente USING (username)
    INNER JOIN datiAnagrafici da USING (username)
    WHERE ordine.data < CURDATE() - 365
    GROUP BY username
    HAVING numOrdine >= 5;
END$$
DELIMITER ;
```

Si vuole ottenere un ordinamento decrescente degli utenti per numero di ordini effettuati nell'ultimo anno

```
DELIMITER $$
CREATE PROCEDURE sp_OrdiniEffettuati()
BEGIN
    SELECT username, nome, cognome, CF, via, città, count(o.numero) as tot
    FROM datiAnagrafici
    INNER JOIN utente u USING (username)
    INNER JOIN ordine o USING (username)
    WHERE o.data > CURDATE() - 365
    GROUP BY username
    ORDER BY tot DESC;
END $$
DELIMITER;
```

Si vuole ottenere un ordinamento decrescente degli utenti per importo pagato nell'ultimo anno

```
DELIMITER $$
CREATE PROCEDURE sp_ImportoPagato()
BEGIN
    SELECT o.username, sum(c.quantità*p.prezzo) as tot FROM prodotto p
    INNER JOIN composizione c USING (codiceProdotto)
    INNER JOIN ordine o USING(numero)
    WHERE o.data > CURDATE()-365
    GROUP BY o.username
    ORDER BY tot DESC;
END $$
DELIMITER;
```

L'username deve essere di almeno 10 caratteri e la password di almeno 8 caratteri

```
CREATE TABLE `utente` (
`username` varchar(30) NOT NULL,
`password` varchar(20) NOT NULL,
`email` varchar(50) NOT NULL,
`telefono` varchar(50) NOT NULL,
PRIMARY KEY (`username`),
CHECK (length(username) >= 10)
CHECK (length(password) >= 8)
);
```

Gli articoli la cui data di produzione risale a più di 2 anni fa non vengono più messi in vendita

```
DELIMITER $$
CREATE TRIGGER trg_controlloDataProdotto
BEFORE INSERT ON prodotto
FOR EACH ROW
BEGIN
    IF new.anno < CURDATE()-730 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Articolo troppo vecchio";
    END IF;
END $$
DELIMITER ;
```