

# **Relazione per il corso di Basi di Dati A.A 2014/2015**

## **InTime The watch market**

**Candidato: Rossetti Alberto  
Matricola: 1050225**

## ABSTRACT

InTime è un sito che si occupa della vendita di orologi da polso, da tasca e di cinturini di qualsiasi marca, prezzo, anno e materiale.

Un sistema basato su un database aggiornabile, nel quale possono essere effettuate ricerche avanzate per poter trovare il segna tempo più adatto ai nostri gusti e al nostro budget, oppure per inserire annunci di vendita in modo rapido e facile.

Il controllo e la trasparenza sono una parte fondamentale per il sito, gli utenti possono mettere in vendita solo orologi conosciuti e presenti nel vasto database, in modo da evitare falsi e avere tutte le informazioni possibili sull'oggetto, se un orologio dovesse mancare nella lista basterà contattare l'amministratore che dopo una valutazione deciderà se aggiungerlo per permettere l'inserzione all'utente.

Gli Utenti iscritti al sito avranno un pannello di gestione degli articoli e la possibilità di effettuare ricerche specifiche e vedere le statistiche InTime, mentre gli utenti ospiti potranno solo vedere le inserzioni senza possibilità di ricerca e di aggiungere articoli.

I negozi sono associati al sito, dopo che un negozio ha effettuato una iscrizione con tutti i dati, il team InTime verifica la loro veridicità instaurando un contatto con il negozio per assicurare la piena affidabilità e professionalità garantendo prezzi scontati e offerte.

Un marketplace che permette ai vari utenti di avere una visione a 360° del commercio di orologi, in modo da poter trovare le migliori offerte e pezzi unici da collezionare, riuscendo ad accontentare qualsiasi tipo di clientela, dal ricercatore di pezzi rari vintage e orologi di lusso, fino all'acquirente di un buon orologio economico.

## DESCRIZIONE DEI REQUISITI

La prima classe è quella delle Persone (utenti) dei quali vengono richiesti Nickname, Password e una classe Profilo che contiene Nome, Cognome, Città di appartenenza, Telefono ed E-mail. Queste classi rappresentano gli utenti iscritti al sito e tutti i campi sono fondamentali e obbligatori per poter iscriversi ad InTime. Ogni Utente può

scrivere, eliminare e modificare più Articoli ed effettuare delle ricerche avanzate tra questi.

Un articolo ha un Id assegnato automaticamente e deve contenere almeno una referenza o un codice di un cinturino esistenti nel DB, un titolo, una descrizione e ovviamente deve essere pubblicato da almeno un Utente o da un Negozio, altre informazioni che si possono ricavare dall'articolo sono la data della pubblicazione, il prezzo che non può superare quello di listino, la condizione da 0 a 10 dell'articolo messo in vendita e la quantità.

La referenza è un codice che identifica univocamente un orologio, come spiegato nell'abstract un articolo può avere solamente un orologio già presente nel database, tutti gli orologi sono contenuti dalla classe Orologi la quale ci offre il numero della referenza, prezzo dell'oggetto nuovo, la marca, il modello, il calibro, il materiale, l'anno, il sesso che può essere M, D o X, il diametro, il tipo che può essere da polso o da tasca, un orologio da polso può essere sia automatico e anche manuale quindi avrà un attributo carica, un orologio da tasca può essere solamente manuale però verrà specificato se è o è stato venduto o meno con una catena da taschino. Il codice cinturino dell'articolo fa riferimento alla classe Cinturini dove sono presenti tutti i codici anche questi univoci come le referenze degli orologi, colori e materiali. Gli articoli possono essere scritti anche da dei Negozi dopo la

solita iscrizione richiesta anche all'utente, dove viene richiesto il Nome, il Nickname, una Password, un Indirizzo, il Telefono e la E-mail. Ogni negozio vende alcune marche di orologi raccolte tutte dalla classe Marche dalla quale si possono ricavare i nomi delle relative firme.

## PROGETTAZIONE CONCETTUALE

### LISTA DELLE ENTITA':

► **PERSONA** : identifica gli utenti iscritti al sito tramite delle credenziali.

- Nickname VARCHAR(20)
- Password VARCHAR(50)

► **PROFILO** : descrive il profilo di ogni utente con tutte le informazioni relative

- Nome VARCHAR(20)
- Cognome VARCHAR(20)
- Città VARCHAR(20)
- Telefono VARCHAR(15)
- Email VARCHAR(30)
- Data\_Iscrizione TIMESTAMP

► **NEGOZIO** : descrive il profilo di ogni negozio

- Nome VARCHAR(20)
- Nickname VARCHAR(20)
- Password VARCHAR(50)
- Indirizzo VARCHAR(30)
- Telefono VARCHAR(15)
- Email VARCHAR(30)

► **MARCHE** : descrive il nome delle marche vendute da ogni negozio

- Nome VARCHAR(20)

► **ARTICOLO** : descrive l'articolo pubblicato da un utente o da un negozio iscritti al sito

- Id INT(4)
- Referenza\_O VARCHAR(20)
- Codice\_C CHAR(3)
- Titolo VARCHAR(50)
- Descrizione VARCHAR(250)
- Prezzo INT
- Condizione ENUM('0','1','2','3','4','5','6','7','8','9','10')

- Quantità INT
- Nickname\_scrittore VARCHAR(20)
- NomeNegozio VARCHAR(20)

► **OROLOGI** : descrive tutte le caratteristiche di ogni singolo orologio riconosciuto dal sito

- Referenza VARCHAR(20)
- Prezzo\_nuovo INT
- Marca VARCHAR(30)
- Modello VARCHAR(30)
- Calibro VARCHAR(10)
- Anno YEAR
- Sesso ENUM('M','D','X')
- Tipo ENUM('tasca','polso')
- *Diametro INT*

E' stata definita la seguente generalizzazione totale ed esclusiva:

► *POLSO* : descrive la carica degli orologi da polso

- Carica ENUM('automatico','manuale','quarzo')

► *TASCA* : ci informa se è presente la catena sull'orologio da polso

- Catena ENUM('presente','non presente')

► **CINTURINI** : descrive tutte le caratteristiche dei cinturini

- Codice CHAR(3)
- Colore VARCHAR(20)
- Materiale VARCHAR(20)

## **LISTA DELLE RELAZIONI:**

► Persona – *Profilo* : **Ha**

- Una persona ha un profilo;
- Un profilo appartiene ad una persona;
- Molteplicità: uno a uno;

► Persona – *Articolo* : **Scrive**

- Una persona scrive 0 o più articoli;
- Un articolo è stato scritto da una persona;
- Molteplicità: uno a molti;

► Cinturini – *Articolo* : **Vende\_C**

- Un articolo ha 0 o 1 cinturino;
- Un cinturino appartiene a 0 o più articoli;
- Molteplicità: uno a molti;

► Negozio – *Articolo* : **Pubblica**

- Un articolo è pubblicato da 0 o un negozio;
- Un negozio pubblica 0 o più articoli;
- Molteplicità: uno a molti;

► Negozio – *Marche* : **Vende**

- Un negozio vende una o più marche;
- Una marca è venduta da uno o più negozi;
- Molteplicità: molti a molti;

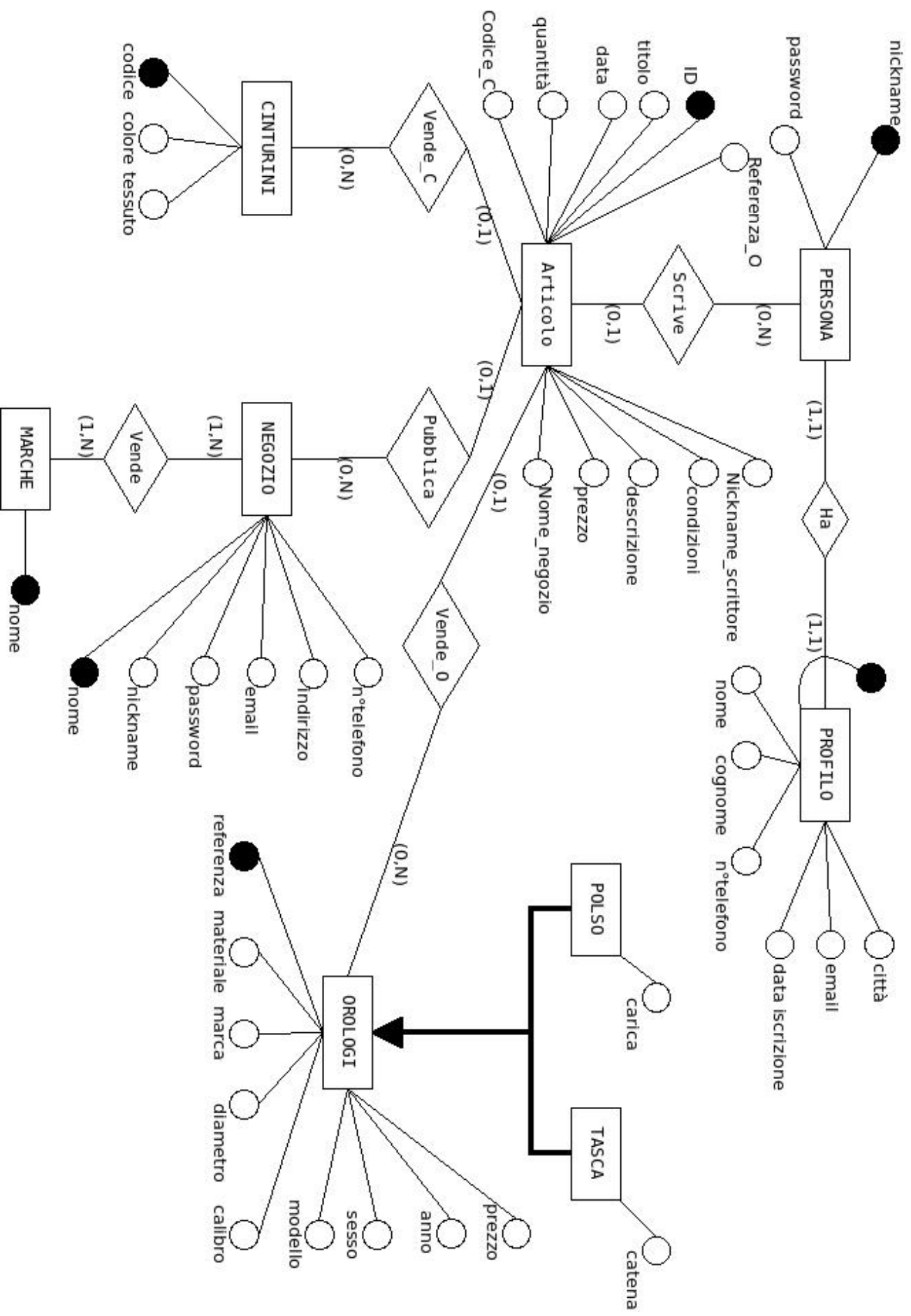
► Articolo – *Orologi* : **Vende\_O**

- Un articolo vende 0 o un orologio;
- Un orologio è venduto da 0 o più articoli;
- Molteplicità: molti a molti;

**LISTA DELLE GENERALIZZAZIONI:**

► Orologi - **Polso/Tasca** :

- *Le entità figlie Polso e Tasca sono esclusive in quanto un Orologio non può essere sia da tasca che da polso dato che hanno caratteristiche diverse. Un orologio da polso può avere una carica automatica, manuale o al quarzo, mentre un orologio da polso può essere solamente manuale però può aver una catenina che prende il posto del bracciale*
- *La generalizzazione è totale poiché ogni occorrenza dell'entità Orologi è una occorrenza di una delle due entità figlie.*



# PROGETTAZIONE LOGICA

## RISTRUTTURAZIONE SCHEMA CONCETTUALE

Per semplificare la traduzione dello schema logico in schema relazionale è necessaria una ristrutturazione dello schema concettuale che permetterà un maggiore supporto dallo schema risultante anche in fatto di ottimizzazione attraverso costrutti supportati dallo schema stesso.

La fase di ristrutturazione si suddividerà in 4 fasi :

- Analisi delle ridondanze. Dove si vedrà se eliminare o mantenere eventuali ridondanze nello schema
- Eliminazione delle generalizzazioni. Tutte le generalizzazioni presenti nello schema verranno analizzate e sostituite da altri costrutti
- Partizionamento/accorpamento di entità e associazioni.
- Scelta degli identificatori principali. Si seleziona un identificatore per quelle entità che ne hanno più di uno

Analisi delle ridondanze: durante la progettazione concettuale è stata fatta molta attenzione a non inserire dati ridondanti per evitare complicazioni nella traduzione allo schema logico.

### Eliminazione delle generalizzazioni:

- La generalizzazione Polso – Tasca – Orologi viene sostituita spostando gli attributi delle entità figlie (Carica,Catena) alle entità genitore. Viene considerata questa scelta dal momento che per le entità figlie ci sono degli attributi che le distinguono in modo preciso l'una dall'altra. Per distinguere se un orologio è da polso o da tasca è stato aggiunto un attributo "Tipo" che ci indica in che categoria si trova l'orologio, a seconda della classe di appartenenza alcuni attributi rimarranno NULL

### Accorpamento di entità e associazioni:

- relazione Ha tra Persona–Profilo : si è deciso di eliminare la relazione e aggregare gli attributi dell'entità profilo all'entità Persona. L'entità Persona quindi avrà come attributi : Nickname,Password,Nome,Cognome,n°telefono,Città,E-mail,Data iscrizione.

In questo modo uniamo il profilo all'utente, questo servirà per semplificare le query ed eliminare una entità diminuendo gli accessi, questa operazione è permessa dato che la molteplicità tra Persona e Profilo è di 1-1.

Si era pensato nella progettazione all'unione tramite una relazione delle due entità Profilo e Negozio, mettendo nell'entità profilo gli attributi in comune tra Negozio e Persona, comunque nella ristrutturazione per arrivare allo schema logico si sarebbe tolta l'entità Profilo che ha una molteplicità 1-1, per evitare lavoro inutile all'entità Negozio sono già state definite le caratteristiche del profilo che hanno leggermente caratteristiche diverse da quelle della Persona.

### Scelta degli identificatori principali:

In questa fase sceglieremo tutte le chiavi di tutte le entità

- ▶ **PERSONA** : la chiave è 'Nickname'.
- ▶ **ARTICOLO** : la chiave è 'Id'.
- ▶ **OROLOGI** : la chiave è 'Referenza'.
- ▶ **CINTURINI** : la chiave è 'Codice'.
- ▶ **NEGOZIO** : la chiave è 'Nome'.
- ▶ **VENDE** : le chiavi sono 'NomeNegozio' e 'NomeMarche'.
- ▶ **MARCHE** : la chiave è 'Nome'.

### SCHEMA RELAZIONALE

- ▶ **PERSONA**(Nickname, Password, Nome, Cognome, Città, Telefono, Email, Data\_Iscrizione)  
*Vincoli di integrità:*
  - la chiave Nickname in Persona e l'attributo Nickname\_Scrittore in Articolo;

- ▶ **ARTICOLO**(Id, Titolo, Data, Quantità, Codice\_C, Referenza\_O, Nickname\_Scrittore, Nome\_Negozio, Descrizione, Prezzo)  
*Vincoli di integrità:*
  - l'attributo Codice\_C in Articolo e l'attributo Codice in Cinturini;
  - l'attributo Referenza\_O in Articolo e l'attributo Referenza in Orologi;
  - l'attributo Nome\_Negozio in Articolo e l'attributo Nome in Negozio;

- ▶ **CINTURINI**(Codice, Colore, Materiale)

- ▶ **NEGOZIO**(Nome, Nickname, Password, Indirizzo, Telefono, Email)  
*Vincoli di integrità:*
  - l'attributo Nome in Negozio e l'attributo Nomenegozio nella relazione Vende\_M;

- ▶ **OROLOGI**(Referenza, Prezzo\_Nuovo, Marca, Modello, Calibro, Carica, Materiale, Anno, Sesso, Diametro, Tipo)

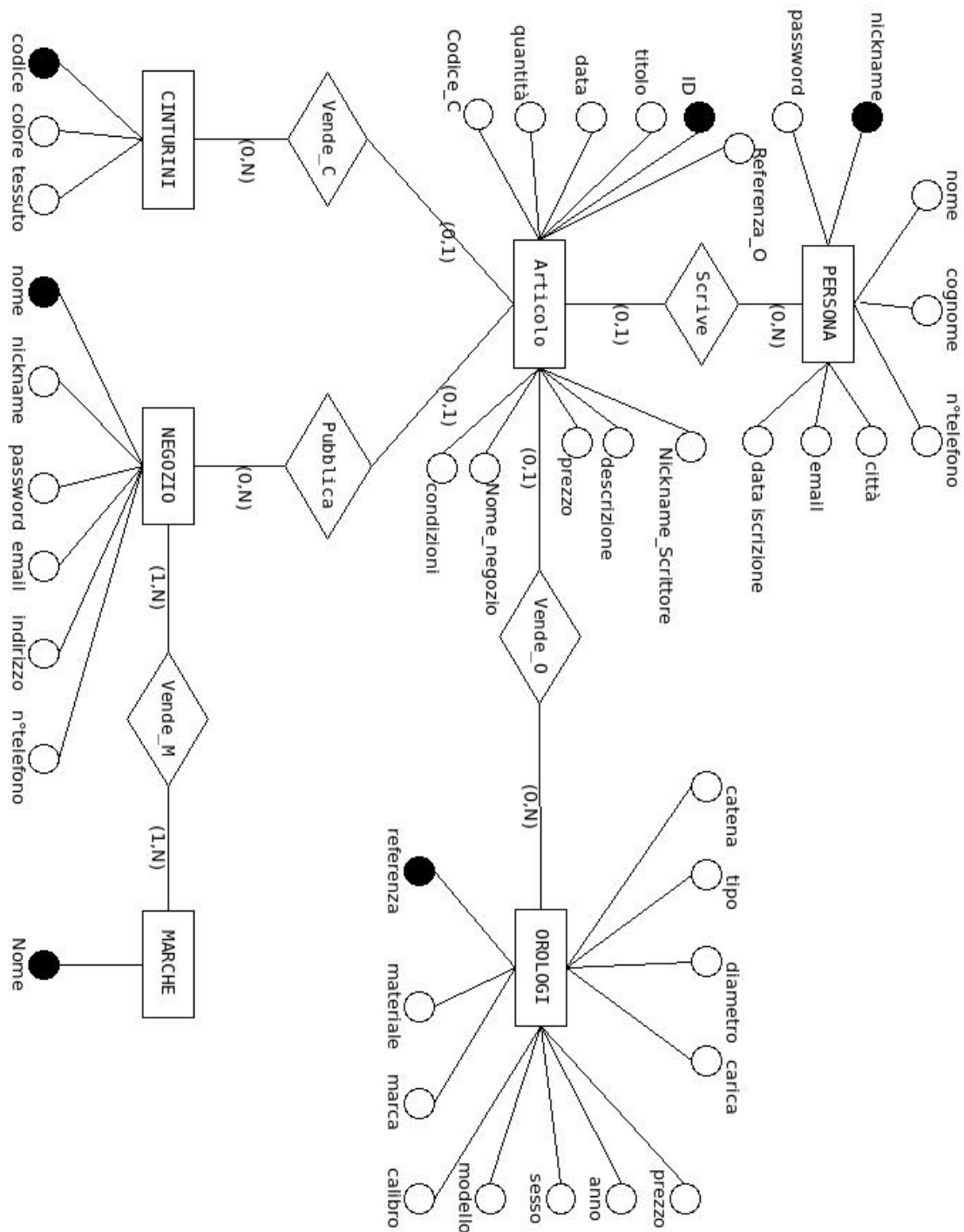


► **VENDE**(NomeNegozio,NomeMarche)

*Vincoli di integrità:*

- l'attributo NomeMarche nella relazione Vende e l'attributo Nome in Marche;

► **MARCHE**(Nome)



## **IMPLEMENTAZIONE BASI DI DATI**

```
SET FOREIGN_KEY_CHECKS=0;
DROP TABLE IF EXISTS Persona;
DROP TABLE IF EXISTS Articolo;
DROP TABLE IF EXISTS Cinturini;
DROP TABLE IF EXISTS Negozio;
DROP TABLE IF EXISTS Orologi;
DROP TABLE IF EXISTS Vende;
DROP TABLE IF EXISTS Marche;
CREATE TABLE Persona (
    Nickname VARCHAR(20) NOT NULL,
    Password VARCHAR(50) NOT NULL,
    Nome VARCHAR(20) NOT NULL,
    Cognome VARCHAR(20) NOT NULL,
    Citta VARCHAR(20),
    Telefono INT(10),
    Email VARCHAR(30) NOT NULL,
    Data_Iscrizione TIMESTAMP default NOW(),
    PRIMARY KEY(Nickname)
)ENGINE=InnoDB;
```

```
CREATE TABLE Articolo (
    Id INT(4) NOT NULL AUTO_INCREMENT,
    Referenza_O VARCHAR(20),
    Codice_C CHAR(3),
    Titolo VARCHAR(50) NOT NULL,
    Data TIMESTAMP default NOW(),
    Descrizione VARCHAR(250) NOT NULL,
    Prezzo INT,
    Condizione ENUM('0','1','2','3','4','5','6','7','8','9','10'),
    Quantita INT ,
    Nickname_scrittore VARCHAR(20),
    NomeNegozio VARCHAR(20),
    PRIMARY KEY(Id),
    FOREIGN KEY (NomeNegozio) REFERENCES Negozio(Nome) ON UPDATE
        cascade ON DELETE set null,
    FOREIGN KEY (Codice_C) REFERENCES Cinturini(Codice) ON UPDATE
        cascade ON DELETE set null,
    FOREIGN KEY (Nickname_scrittore) REFERENCES Persona(Nickname) ON
        UPDATE cascade ON DELETE set null,
    FOREIGN KEY (Referenza_O) REFERENCES Orologi(Referenza) ON UPDATE
        cascade ON DELETE set null
) ENGINE=InnoDB;
```

```
CREATE TABLE Cinturini (
    Codice CHAR(3) NOT NULL,
    Colore VARCHAR(20) NOT NULL,
    Materiale VARCHAR(20) NOT NULL,
    PRIMARY KEY(Codice)
)ENGINE=InnoDB;
```

```
CREATE TABLE Negozio (
    Nome VARCHAR(20) NOT NULL,
    Nickname VARCHAR(20) UNIQUE NOT NULL,
    Password VARCHAR(50) NOT NULL,
    Indirizzo VARCHAR(30) NOT NULL,
    Telefono INT(13) NOT NULL,
    Email VARCHAR(30) NOT NULL,
    PRIMARY KEY(Nome)
)ENGINE=InnoDB;
```

```
CREATE TABLE Orologi (
    Referenza VARCHAR(20) NOT NULL,
    Prezzo_nuovo INT NOT NULL,
    Marca VARCHAR(30) NOT NULL,
    Modello VARCHAR(30) NOT NULL,
    Calibro VARCHAR(10) NOT NULL,
    Carica ENUM('automatico','manuale','quarzo'),
    Materiale VARCHAR(20),
    Anno YEAR,
    Sesso ENUM('M','D','X'),
    Diametro INT,
    Catena ENUM('presente','non presente'),
    Tipo ENUM('tasca','polso'),
    PRIMARY KEY(Referenza)
)ENGINE=InnoDB;
```

```
CREATE TABLE Vende (
    Nomenegozio VARCHAR(20),
    NomeMarche VARCHAR(20),
    PRIMARY KEY(Nomenegozio,NomeMarche),
    FOREIGN KEY (Nomenegozio) REFERENCES Negozio(Nome) ON UPDATE
cascade ON DELETE cascade,
    FOREIGN KEY (NomeMarche) REFERENCES Marche(Nome) ON UPDATE
cascade ON DELETE cascade
)ENGINE=InnoDB;
CREATE TABLE Marche (
    Nome VARCHAR(20),
    PRIMARY KEY(Nome)
)ENGINE=InnoDB;
```

## Query, Trigger e Funzioni

### Query:

1) Visualizza gli articoli caricati dal 2014 al 2015 che hanno una condizione maggiore di 8, di sesso M, di marca con iniziale 'O', ordinati per prezzo decrescente.

```
SELECT a.Titolo,a.Referenza_O,o.Prezzo_nuovo,a.Prezzo,o.Marca,o.Calibro,
        o.Sesso,o.Carica,a.Nickname_scrittore,a.NomeNegozio,a.Data
FROM Articolo a JOIN Orologi o ON (a.Referenza_O=o.Referenza)
WHERE YEAR(a.Data)>='2014' AND YEAR(a.Data)<'2015' AND o.Sesso='M' AND
        o.Marca LIKE 'O%' AND a.Condizione>8
ORDER BY a.Prezzo DESC
```

<u>Titolo</u>	<u>Referenza</u>	<u>Prezzo</u> <u>Nuovo</u>	<u>Prezzo</u>	<u>Marca</u>	<u>Calibro</u>	<u>Sesso</u>	<u>Carica</u>	<u>Nickname</u> <u>Scrittore</u>	<u>Nome</u> <u>Negozio</u>	<u>Data</u>
vendo Omega speedmaster professional	<a href="#">239821</a>	9000€	4700€	Omega	p701	M	Automatica	NULL	<a href="#">Brevet</a>	2014-03-22
vendo Omega planetocean	<a href="#">549811</a>	10000€	4300€	Omega	p708	M	Automatica	NULL	<a href="#">Brevet</a>	2014-07-22
vendo Omega 300M professional	<a href="#">438147</a>	8000€	2400€	Omega	p700	M	Automatica	NULL	<a href="#">Brevet</a>	2014-03-22

2) Articoli che hanno uno sconto superiore al 80% ordinati per prezzo decrescente, con una marca venduta da uno dei negozi associati

```
SELECT a.Id,a.Prezzo,o.Prezzo_nuovo,a.Referenza_O,o.Marca,a.Titolo,
        a.Descrizione,a.Nickname_Scrittore,a.NomeNegozio
FROM Articolo a JOIN Orologi o ON a.Referenza_O=o.Referenza
WHERE (o.Prezzo_nuovo-a.Prezzo)>=((o.Prezzo_nuovo*80)/100)
        AND o.Marca IN (SELECT Nome
                        FROM Marche)
ORDER BY a.Prezzo DESC
```

<u>Id</u>	<u>Prezzo</u>	<u>Prezzo</u> <u>Nuovo</u>	<u>Referenza</u> <u>O</u>	<u>Marca</u>	<u>Titolo</u>	<u>Descrizione</u>	<u>Nickname S</u> <u>crittore</u>	<u>Nome</u> <u>Negozio</u>
38	3000€	18000€	<a href="#">1123</a>	Patek	vendo patek aquanaut 2015	NOS 2014 con cinturini in gomma nero	NULL	<a href="#">Ferrari</a>
4	1400€	9000€	<a href="#">2523</a>	Rolex	vendo Rolex Datejust anni70 perfetto	date just anni 70 senza corredo ma in condizioni e...	filip	NULL
21	1200€	8000€	<a href="#">2231847</a>	Zenith	vendo Zenith Elite	zenith elite NOS 2015	NULL	<a href="#">Brevet</a>
7	220€	1200€	<a href="#">21397</a>	Longines	vendo Longines Hydroconquest L3	orologio sub satinato,ottimo meccanismo affidabile...	GeoP	NULL

3) Di tutti gli orologi Omega trovare tutti i rispettivi articoli con un prezzo sotto i 7000€ visualizzando chi li ha messi in vendita con anno dell'orologio successivo al 2000 ordinati per data di inserimento articolo in ordine decrescente

```
SELECT o.Marca,o.Modello,o.Calibro,o.Carica,o.Referenza,o.Anno,a.Id,
       a.Titolo,a.NomeNegozio,n.Email
FROM (Articolo a JOIN Negozio n ON a.NomeNegozio=n.Nome)
      JOIN Orologi o ON a.Referenza_O=o.Referenza
WHERE a.Prezzo<7000 AND o.Marca='Omega' AND o.Anno>'2000'
ORDER BY a.Data DESC
```

Marca	Modello	Calibro	Carica	Referenza	Anno	Id	Titolo	Nome Negozio	Email
Omega	planetocean	p708	automatico	549811	2015	24	vendo Omega planetocean	<a href="#">Brevet</a>	Brevet@gmail.com
Omega	Speedmaster	p701	automatico	239821	2015	23	vendo Omega speedmaster professional	<a href="#">Brevet</a>	Brevet@gmail.com
Omega	300M	p700	automatico	438147	2015	22	vendo Omega 300M professional	<a href="#">Brevet</a>	Brevet@gmail.com

4) Trovare articoli marca Zenith con cinturino in acciaio del Negozio Brevet con Nickname "Bre67". Per semplificare questa query è stata creata una vista.

```
CREATE VIEW articolinegozi AS
SELECT a.Id AS Id,a.Referenza_O AS Referenza,a.Prezzo AS Prezzo,o.Marca AS
       Marca,o.Modello AS Modello,c.Codice AS Codice_c,c.Materiale AS
       Materiale, a.NomeNegozio AS NomeNegozio,n.Email AS Email
FROM ((Articolo a JOIN Negozio n ON a.NomeNegozio=n.Nome)
      JOIN Cinturini c ON a.Codice_C=c.Codice)
      JOIN Orologi o ON a.Referenza_O=o.Referenza
WHERE o.Marca='Zenith' AND c.Materiale='acciaio'
```

```
SELECT Id ,Referenza ,Marca ,Modello ,Prezzo ,Codice_C ,Materiale ,Email
FROM articolinegozi
WHERE NomeNegozio=(SELECT Nome
                    FROM Negozio
                    WHERE Nickname='Bre67')
```

Id	Referenza	Marca	Modello	Prezzo	Codice_C	Materiale	Email
41	35423	Zenith	Stellina	300€	635	acciaio	Brevet@gmail.com

- 5) Cercare il negozio che ha il maggior numero di orologi disponibili in vendita.  
Per semplificare questa query è stata creata una vista.

```
CREATE VIEW narticoli AS
SELECT DISTINCT Nome,SUM(Quantita) AS Sommaquantita
FROM (Articolo JOIN Negozio ON Nome=NomeNegozio)
GROUP BY Nome
ORDER BY Sommaquantita desc
```

```
SELECT Nome, MAX(Sommaquantita) AS TotaleMassimo
FROM narticoli
```

Nome	TotaleMassimo
LaRocca	34

- 6) Visualizzare l'Orologio più caro con marca riconosciuta dal sito

```
SELECT a.Id,a.Prezzo,a.Condizione,a.Riferenza_O,o.Marca,o.Modello,o.Calibro,
a.Titolo,a.Descrizione,a.Nickname_Scrittore,a.NomeNegozio
FROM Articolo a JOIN Orologi o ON a.Riferenza_O=o.Riferenza
WHERE o.Marca IN (SELECT Nome
FROM Marche)
AND
a.Prezzo=(SELECT MAX(Prezzo)
FROM Articolo)
```

<u>Id</u>	<u>Prezzo</u>	<u>Condizione</u>	<u>Riferenza_O</u>	<u>Marca</u>	<u>Modello</u>	<u>Calibro</u>	<u>Titolo</u>	<u>Descrizione</u>	<u>Nickname_Scrittore</u>	<u>NomeNegozio</u>
26	35000€	9	<a href="#">123</a>	Rolex	PaulNewman	R543	vendo Rolex PaulNewman	PaulNewman perfetto e raro,corredo completo e gara...	NULL	<a href="#">Tempus</a>

- 7) Trovare la media delle condizioni e il numero degli articoli totali con marche riconosciute al sito

```
SELECT Id,avg(Condizione-1)
FROM Articolo a JOIN Orologi o ON a.Riferenza_O=o.Riferenza
WHERE o.Marca IN (SELECT NomeMarche
FROM Vende)
```

<u>Id</u>	<u>Media Condizioni</u>
45	9.31578947368421

## Funzioni:

1) Funzione che ritorna il numero degli articoli messi in vendita quest'ultimo mese.

```
DELIMITER //
CREATE FUNCTION ultimiorologi() RETURNS INT
BEGIN
    DECLARE conta INT DEFAULT 0;
    SELECT COUNT(*) INTO conta
    FROM Articolo
    WHERE DATEDIFF(NOW(),Data)<30;
    RETURN conta;
END;
```

2) Procedura per inserire il cinturino con codice passato per valore nell'articolo con Id e Nickname passati anche essi per valore.

```
DELIMITER //
CREATE PROCEDURE cinturino (IN Id_art INT(4),IN Cinturino CHAR(3),IN
                             Nickname VARCHAR(20))
BEGIN
    DECLARE cod CHAR(3);
    SELECT Codice_C INTO cod
    FROM Articolo
    WHERE Id=Id_art;
    IF cod IS NULL THEN
        UPDATE Articolo
        SET Codice_C=Cinturino
        WHERE Nickname_Scrittore = Nickname AND Id=Id_art;
    END IF;
END;
```

3) Funzione che ritorna la data dell'ultimo articolo pubblicato dall'utente con Nickname passato per valore.

```
DELIMITER //
CREATE FUNCTION tempo(Nickname VARCHAR(20)) RETURNS TIMESTAMP
BEGIN
    DECLARE dataultimoart TIMESTAMP;
    SELECT MAX(Data) INTO dataultimoart
    FROM Articolo
    WHERE Nickname_Scrittore = Nickname;
    RETURN dataultimoart;
END;
```

## Trigger:

1) Questo trigger si attiva quando facciamo un'iscrizione di un negozio, controlla che tutti i campi obbligatori siano compilati, in caso negativo impedisce l'inserimento

```
CREATE TRIGGER ControllonegoziINS
BEFORE INSERT ON Negozio
FOR EACH ROW
BEGIN
    IF (NEW.Nickname = '' OR NEW.Password = MD5('') OR New.Nome = '' OR
        New.Indirizzo = '' OR New.Telefono = '' OR New.Email = '') THEN
        CALL `compilare i campi`;
    END IF;
END $$
```

2) Questo trigger viene attivato quando andiamo a modificare la Password dopo aver effettuato la login come Negozio, in modo specifico controlla che la password vecchia **non** sia uguale a quella nuova

```
CREATE TRIGGER ControllonegoziUPD
BEFORE UPDATE ON Negozio
FOR EACH ROW
BEGIN
    IF (NEW.Password = OLD.Password) THEN
        CALL `psw vecchia uguale alla nuova`;
    END IF;
END $$
```

3) Questo trigger ha la stessa funzione del trigger n°1 però il controllo viene fatto al livello della tabella Persona (Utente)

```
CREATE TRIGGER ControlloiscrizioneINS
BEFORE INSERT ON Persona
FOR EACH ROW
BEGIN
    IF (NEW.Nickname = '' OR NEW.Password = MD5('') OR New.Nome = '' OR
        New.Cognome = '' OR New.Email = '') THEN
        CALL `compilare i campi`;
    END IF;
END $$
```

4) Questo trigger ha la stessa funzione del trigger n°2 però il controllo viene fatto al livello della tabella Persona (Utente)

```
CREATE TRIGGER ControllopswUPD
BEFORE UPDATE ON Persona
FOR EACH ROW
BEGIN
    IF (NEW.Password = OLD.Password) THEN
        CALL `psw vecchia uguale alla nuova`;
    END IF;
END $$
```



5) Questo trigger fa un doppio controllo in Update sulla tabella Orologi. Verifica che il diametro immesso non sia >48 (massimo diametro per un orologio) e che un orologio da tasca non può avere una carica automatica o al quarzo, in questo modo viene controllata la generalizzazione fatta nello schema ER , lo stesso trigger è stato fatto anche in inserimento.

```
CREATE TRIGGER ControlloOrologioUPD
BEFORE UPDATE ON Orologi
FOR EACH ROW
BEGIN
    IF(New.Diametro > 48) THEN
        CALL `errore diametro`;
    END IF;
    IF(New.Tipo = "tasca" AND (New.Carica = "automatico" OR
        New.Carica = "quarzo")) THEN
        CALL `controllare tipo e carica`;
    END IF;
    IF(New.Tipo = "polso") THEN
        SET New.Catena = "non presente";
    END IF;
END $$
```

6) Questo trigger si attiva prima di fare un inserimento nella cartella articolo, quando omettiamo la quantità la setta di default a 1 , controlla che sia inserita almeno una referenza di un orologio o un codice di un cinturino, controlla che siano inseriti un titolo e una descrizione e se viene immessa una referenza di un orologio, controlla che il prezzo dell'articolo non sia maggiore del prezzo di listino, controlla che se viene inserito un orologio da tasca non può essere inserito un cinturino. Questo trigger esiste simile anche in update.

```
CREATE TRIGGER ControlloArticoloINS
BEFORE INSERT ON Articolo
FOR EACH ROW
BEGIN
    IF(New.Quantita = '') THEN
        SET New.Quantita=1;
    END IF;
    IF(New.Codice_C = '' AND New.Referenza_O = '') THEN
        CALL `errore articolo`;
    END IF;
    IF(New.Titolo = '' OR New.Descrizione = '') THEN
        CALL `errore articolo`;
    END IF;
    IF(New.Codice_C IS NOT NULL
        AND New.Referenza_O = (SELECT Orologi.Referenza
                                FROM Orologi
                                WHERE Orologi.Referenza=
                                    New.Referenza_O AND Orologi.Tipo="tasca")) THEN
        CALL `errore articolo`;
    END IF;
    IF(New.Referenza_O IS NOT NULL) THEN
        IF(New.Prezzo > (SELECT Orologi.Prezzo_nuovo FROM Orologi WHERE
                                Orologi.Referenza=New.Referenza_O)) THEN
            CALL `errore prezzi`;
        END IF;
    END IF;
END
```

```
END IF;  
END IF;  
END $$
```

## **INTERFACCIA WEB**

L'interfaccia web si presenta contenuta all'interno di un sito di presentazione per un portale di vendita di orologi.

La possibilità di accedere alle varie funzionalità della base di dati si ottiene passando l'autenticazione della pagina login.php.

L'autenticazione è realizzata tramite la memorizzazione in variabili di sessione del tipo Nickname e Persona/negozio dopo aver controllato nella base di dati se la password utilizzata è corretta per l'utente/negozio che vuole accedere alla pagina. Ogni pagina controlla il tipo di login e in base a questo visualizzerà le varie informazioni, altrimenti restituirà una pagina di errore con un indirizzamento verso la login.

Un utente "ospite" potrà avere delle credenziali di login dopo aver effettuato una iscrizione ad InTimes nella apposita sezione(iscrizione utente/negozio) dove saranno richiesti un Nickname (nel quale ci sarà un controllo, se il Nickname è già esistente verrà segnalato un errore),una Password,un Nome,un Cognome,la Città (non obbligatoria),un numero di Telefono(non obbligatoria) e una E-mail. L'iscrizione per un Negozio chiederà delle credenziali leggermente diverse.

Dopo aver effettuato l'iscrizione ci sarà un refresh pagina che ci farà accedere automaticamente al sito senza passare per la pagina login.php.

Qualora l'utente/negozio voglia abbandonare la sessione basterà cliccare su Logout.

### **Autenticazione come Utente o Negozio**

Una Persona/Negozio autenticata come Utente, potrà accedere alla pagina miei\_articoli.php, nella quale verranno visualizzati tutti gli articoli scritti da esso e un pannello per gestirli (modifica,aggiunta ed eliminazione) modifica.php e modifica2.php .

Inoltre dopo il Login si potranno effettuare delle ricerche avanzate all'interno dell'area 'Articoli' , si potrà cambiare la Password e si potrà prendere visione delle statistiche del sito oltre alle normali funzionalità che avrebbe un Utente ospite.

Se il Login non sarà effettuato tutte le pagine in più disponibili per gli utenti iscritti non saranno accessibili per un utente ospite, se tale proverà ad accedere alle aree esclusive agli iscritti si presenterà un messaggio d'errore con conseguente reindirizzamento alla Login.

## **Interfaccia comune**

Tutte le pagine presentano un menù di navigazione principale che permette di orientarsi nel sito InTimes, se viene visitato come “Ospite” avremo un accesso alle pagine :

index.php,articoli.php,negozi.php,utenti.php,login.php,iscrizione\_utente.php,iscrizione\_negozio.php, se accediamo al sito avremo in più la sezione miei\_articoli.php,logout.php,change-pwd e statistiche\_sito.php.

I passaggi dei parametri tra le pagine vengono realizzati tramite variabili di sessione per l'autenticazione, che sblocca/blocca le varie funzionalità.

Per semplicità indico quattro utenti con cui è possibile testare l'interfaccia web:

Utente 1:

- Nickname : rachel
- Password : powerbit

Utente 2:

- Nickname : rickyb93
- Password : TheRock

Negozio 1:

- Nickname : Bre67
- Password : Bre76

Negozio 2:

- Nickname : Tempus\_PD
- Password : Romex222

## **Accesso Database**

L'accesso al database tramite phpMyAdmin ha credenziali :

Username : arossett

Password : 0eJGrPFn