



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Progetto BDSI 2021/2022
Realizzazione di una base di dati di uno
shop per articoli NVIDIA

Realizzato da:

Morandini Giulio, 7030209

Tafazoli Aria, 7012975

Leone Tommaso, 7052301

Richiesta

Si vuole realizzare la base di dati PCSHOP dell'azienda NVIDIA, che vende componenti pc. Ogni cliente è identificato univocamente da un ID Utente, ha rispettivamente un nome e un cognome, un indirizzo di spedizione, , una e-mail, un recapito telefonico ed un eventuale numero di partita IVA.

I clienti possono acquistare i prodotti disponibili.

Ogni prodotto è identificato da un codice, ha un nome, un prezzo, una categoria e il suo anno di produzione.

La fase di acquisto è caratterizzata dall'inserimento dei prodotti nel carrello, che è unico per ogni cliente.

Di ogni carrello conosciamo la quantità dei prodotti ed il costo finale per i prodotti inseriti.

Alla conferma dell'ordine viene emessa una fattura con un suo numero di fattura e l'elenco dei prodotti, con i rispettivi prezzi e la data di emissione.

Ciascun ordine viene salvato con il suo numero d'ordine, il cliente che l'ha effettuato, la data in cui è stato effettuato, la fattura associata e delle spese di spedizione.

Progettazione concettuale

La traduzione delle specifiche in schema ER è avvenuta attraverso una strategia top-down.

È stata considerata l'entità **Prodotto**, con chiave primaria **IDProdotto** (1,1). Sono stati aggiunti anche gli attributi **nome** (1,1), **prezzo** (1,1), **categoria** (1,1), **anno** (1,1).

È stata aggiunta l'entità **Carrello**, identificata dall'attributo **IDCarrello** (1,1) e dall'attributo **quantità** (1,1), ovvero la quantità di prodotti presenti nel carrello.

È stata considerata l'entità **Ordine**, con gli attributi **numero dell'ordine** (1,1) come l'identificatore, **data dell'ordine** (1,1), **spese di spedizione** (1,1).

È stata creata l'entità **Cliente** identificata dall'attributo **IDUtente** (1,1), con caratteristiche **nome** (1,1), **cognome** (1,1), **indirizzo** (1,1), **indirizzo mail** (1,1), **recapito telefonico** (1,1) ed una eventuale **partita IVA**(0,1) come attributo opzionale.

Dopo la conferma di un ordine viene emessa una **Fattura**, definita da una sua entità con chiave primaria **IDFattura** (1,1).

È stata aggiunta la relazione molti a molti **inserimento** tra *prodotto e carrello*: un prodotto può essere inserito in nessuno o più carrelli ed un carrello può essere vuoto o avere tanti prodotti.

Un ordine può essere *in corso*, oppure *concluso*, pertanto è stata aggiunta la generalizzazione *totale ed esclusiva* degli ordini in **Ordine In Corso** ed **Ordine Concluso**.

È stata aggiunta la relazione uno a uno **collegamento** tra *carrello e ordine in corso*.

È stata aggiunta la relazione **effettuazione** tra il *cliente* e l'*ordine in corso*. Visto che un cliente può avere zero o tanti ordini in corso ed un ordine in corso può avere uno ed un solo cliente la relazione è stata definita con cardinalità uno a molti.

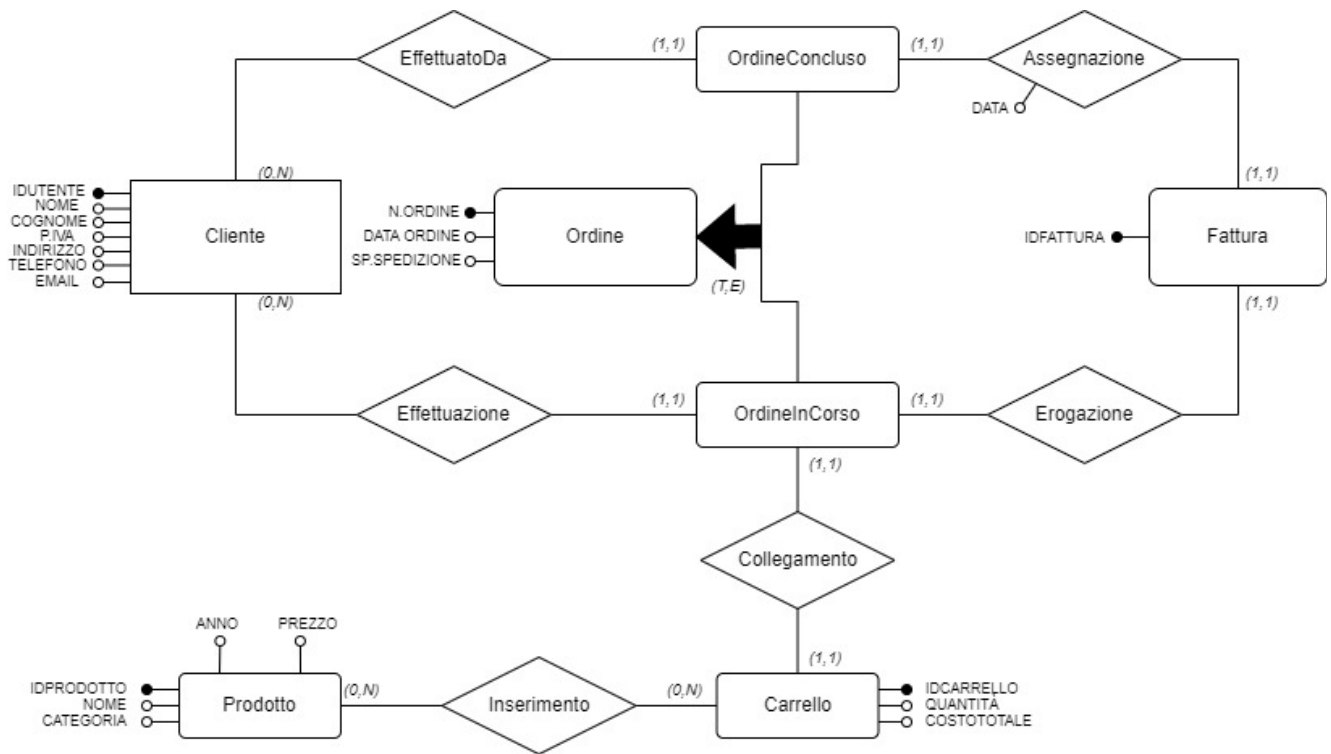
È stata aggiunta la relazione **effettuato da** tra l'*ordine concluso* ed il *cliente*. Visto che un cliente può avere zero o tanti ordini effettuati ed un ordine concluso può avere uno ed un solo cliente la relazione è stata definita con cardinalità uno a molti.

È stata aggiunta la relazione **erogazione** tra il *fattura* e *ordine in corso*. Dato che un ordine ha una ed una sola fattura associata, e una fattura è associata solo ad un ordine, la relazione è definita con cardinalità uno a uno

È stata aggiunta la relazione **assegnazione** tra *ordine concluso* e *fattura*. Una fattura è assegnata ad un solo ordine, e un ordine ha una ed una sola fattura associata, quindi la relazione avrà cardinalità uno a uno.

Viene attribuita poi la **data di fattura**.

Schema Entità-Relazione



Dizionario dei dati – Entità

Entità	Descrizione	Attributi	Chiave primaria
Cliente	Utenti registrati nel sito	IDUtente, nome, cognome, email, indirizzo, partitaIva, telefono	IDUtente
Prodotto	Acquistabile dallo shop	IDProdotto, nome, categoria, anno, prezzo	IDProdotto
Ordine	Cosa vogliono acquistare i clienti	numOrdine, dataOrdine, speseSpedizione	numOrdine
OrdineInCorso	Ordine del cliente attualmente in corso		
OrdineConcluso	Ordine del cliente concluso e archiviato		
Carrello	Lista associata ad ogni utente contenente tutti i prodotti che esso vuole acquistare	IDCarrello, quantità, costoTotale	IDCarrello
Fattura	la ricevuta associata ad ogni ordine contenente informazioni su di esso	IDFattura, dataFattura	IDFattura

Dizionario dei dati – Relazioni

Relazioni	Componenti	Attributi
EffettuatoDa	Cliente, OrdineConcluso	
Effettuazione	Cliente, OrdineInCorso	
Assegnazione	OrdineConcluso, Fattura	dataFattura
Erogazione	OrdineInCorso, Fattura	
Collegamento	OrdineInCorso, Carrello	
Inserimento	Prodotto, Carrello	

Progettazione logica

Analisi delle prestazioni

Tavola dei volumi

Concetto	Tipo	Volume
Cliente	E	500
Ordine	E (generalizzazione)	2000
Ordine Concluso	E (specializzazione)	2000
Ordine In Corso	E (specializzazione)	200
Prodotto	E	1000
Carrello	E	70
Fattura	E	2000
Effettuato da	R	2000
Effettuazione	R	2000
Assegnazione	R	2000
Erogazione	R	2000
Collegamento	R	2000
Inserimento	R	5000

Tavola delle operazioni

Operazione	Frequenza
Operazione 1 - Scegliere dei prodotti e inserirli nel carrello	5 per cliente
Operazione 2 - Acquisizione	20 clienti al giorno
Operazione 3 – Erogazione ordine e fattura	1 per cliente

Tavola degli accessi

Operazione 1 - Scegliere dei prodotti e inserirli nel carrello:

Concetto	Costrutto	Accessi	Tipo
Prodotto	E	5	Scrittura
Inserimento	R	5	Scrittura
Carrello	E	5	Scrittura

Operazione 2 - Acquisizione:

Concetto	Costrutto	Accessi	Tipo
Cliente	E	1	Scrittura
Effettuazione	R	1	Scrittura
Ordine in corso	E	1	Scrittura
Collegamento	R	1	Scrittura
Carrello	E	1	Scrittura

Operazione 3 - Erogazione ordine e fattura

Concetto	Costrutto	Accessi	Tipo
Cliente	E	1	Lettura
Effettuato da	R	1	Lettura
Ordine concluso	E	1	Scrittura
Assegnazione	R	1	Scrittura
Fattura	E	1	Scrittura

Analisi delle ridondanze

Possiamo prendere in analisi le due date, ovvero quella dell'ordine e quella della fattura dato che probabilmente potrebbero indicare la stessa informazione; potrebbe però capitare che le due date non coincidano, quindi dobbiamo mantenere entrambi gli attributi.

Potremo anche eliminare gli attributi quantità e costoTotale nell'entità Carrello, dato che in algebra relazionale esistono comunque delle operazioni che ci permettono di calcolarle, ovvero *sum* e *count*. Abbiamo deciso però di lasciarli per evitare di doverli ricalcolare ogni volta che si esegue un'interrogazione.

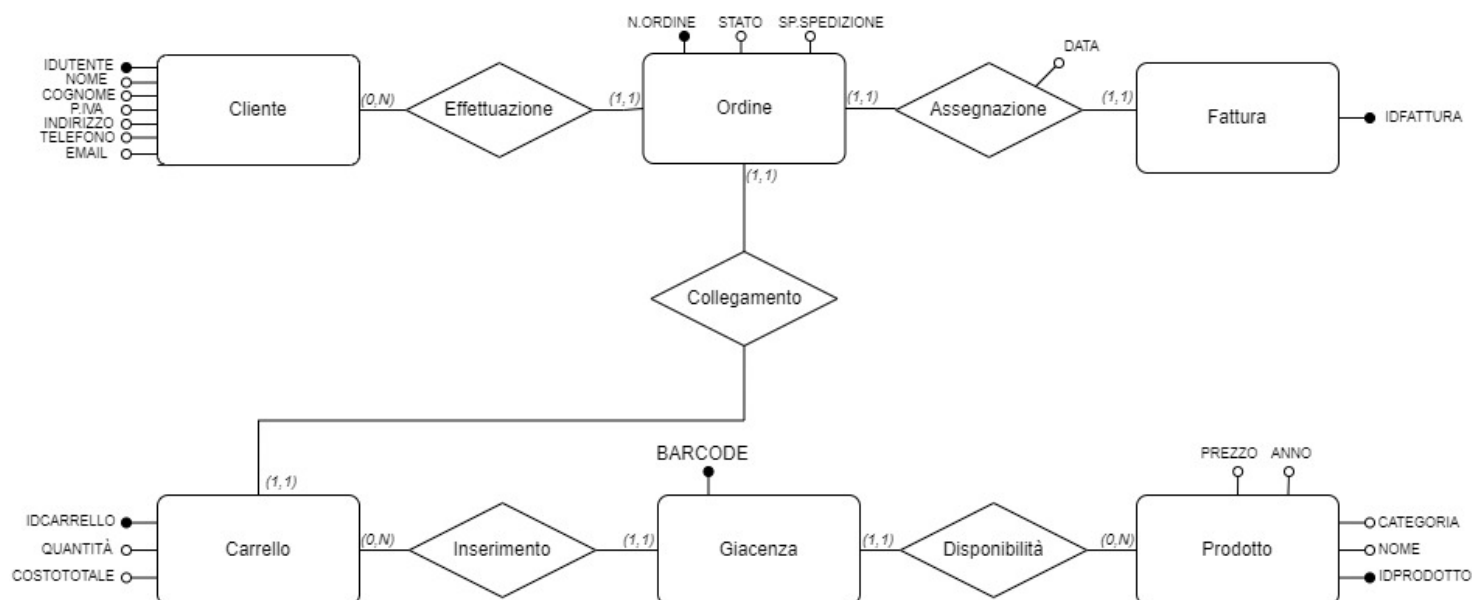
Eliminazione delle generalizzazioni

In questa fase si eliminano tutte le generalizzazioni esistenti dato che esse non sono rappresentabili nel modello logico. Visto che abbiamo una generalizzazione (totale ed esclusiva) ma per accedere alle figlie, ovvero *OrdineInCorso* e *Ordine concluso*, non facciamo distinzione, abbiamo scelto di applicare ***l'accorpamento delle figlie al genitore***. Quindi eliminamo la generalizzazione portando gli attributi delle entità figlie nel genitore, aggiungendo anche un altro attributo "stato" per poter distinguere il tipo di occorrenza dell'ordine, cioè se *in corso* o *concluso*.

Quindi dopo la generalizzazione anche le associazioni EffettuatiDa ed Erogazione vengono rispettivamente unite con le associazioni Effettuazione ed Assegnazione.

È stata aggiunta infine una entità *giacenza*, per semplificare la relazione inserimento dato che è una relazione di tipo molti a molti.

Ristrutturazione finale dello schema ER



Progettazione logica

Conversione del diagramma E-R a modello relazionale

Il risultato finale è il seguente, ottenuto tramite l'applicazione di regole banali di traduzione:

Cliente (IDUtente, nome, cognome, email, indirizzo, partitaIva, telefono)

Prodotto (IDProdotto, nome, categoria, anno, prezzo)

Giacenza (barcode, IDProdotto)

Giacenza.IDProdotto → Prodotto.IDProdotto

Carrello (IDCarrello, quantità, costoTotale)

Ordine (numOrdine, dataOrdine, speseSpedizione, stato, IDCarrello)

OrdineInCorso.IDCarrello → Carrello.IDCarrello

Fattura (IDFattura, dataFattura, numOrdine)

Fattura.numOrdine → Ordine.numOrdine

Effettuazione (numOrdine, IDUtente)

Effettuazione.numOrdine → Ordine.numOrdine

Effettuazione.IDUtente → Cliente.IDUtente

Inserimento (barcode, IDCarrello)

Inserimento.barcode → Giacenza.barcode

Inserimento.IDCarrello → Carrello.IDCarrello

Codice SQL

Per i dettagli della stesura del codice SQL, si rimanda direttamente al codice.
Il codice è stato suddiviso seguendo il template fornito, suddividendolo in:

- **CREAZIONE DATABASE:** dove viene creato il database.
- **CREAZIONE TABELLE:** dove vengono create le varie entità necessarie, ovvero *fattura*, *ordine*, *carrello*, *giacenza*, *prodotto* e *cliente*, e le relazioni *inserimento* ed *effettuazione*.
- **OPERAZIONI A LIVELLO DI SCHEMA:** vengono descritte due operazioni, la prima che modifica il prezzo possibile del carrello, rendendolo più capiente, ovvero da sei a sette cifre intere, e la seconda che aggiunge *costoTotale* a *carrello*.
- **VISTA:** vista degli ordini in corso.
- **FUNZIONE:** funzione che restituisce il numero di ordini per cliente.
- **PROCEDURA:** vengono descritte due procedure, una per calcolare il costo totale ed un'altra per eventualmente togliere l'iva.
- **TRIGGER:** dove sono presenti i vari trigger, ovvero il primo calcola la quantità di prodotti nel carrello ad ogni inserimento di un nuovo prodotto, il secondo aggiorna il carrello ogni volta che viene aggiunto un nuovo articolo, modificando costo totale ed il terzo toglie l'iva.
- **POPOLAMENTO DATABASE:** dove viene popolata la base di dati.
- **QUERY:** dove sono presenti tre query descritte direttamente nel codice.