**Caso di Studio: E-commerce di Prodotti Cosmetici**

**Descrizione del Caso di Studio:**

Nel caso di studio scelto stiamo creando una base dati per un'azienda che vende prodotti cosmetici online.

L'azienda ha un’ampia gamma di prodotti, dai trucchi alle creme per la cura della pelle.

Gli utenti possono sfogliare il catalogo, aggiungere prodotti al carrello ed effettuare acquisti online.

L'azienda dispone di dipendenti che si occupano degli ordini, del magazzino e del supporto clienti.

**Entità:**

**Utenti**

Attributi: IDUtente (PK), Nome, Cognome, Email, DataRegistrazione

**Prodotti**

Attributi: IDProdotto (PK), Nome, Descrizione, Prezzo

**Ordini**

Attributi: IDOrdine (PK), IDUtente (FK), IDDipendente(FK), DataOrdine, Totale

Relazioni:

Chiavi esterne: IDUtente fa riferimento a Utente(IDUtente) e IDDipendente fa riferimento a Dipendente(IDDipendente)

**DettaglioOrdine**

Attributi: IDDettaglio (PK), IDOrdine (FK), IDProdotto (FK), Quantità

Relazioni:

Chiavi esterne: IDOrdine fa riferimento a Ordine(IDOrdine), IDProdotto fa riferimento a

Prodotto(IDProdotto)

**Dipendenti**

Attributi: IDDipendente (PK), Nome, Cognome, Ruolo

**IndirizzoSpedizioni**

Attributi: IDIndirizzo(PK), Via, Città, Cap, Paese

**UtenteIndirizzo**

Attributi: IDUtenteIndirizzo(PK), IDUtente(FK), IDIndirizzo(FK)

Chiavi esterne: IDUtente che fa riferimento a Utenti(IDUtente), IDIndirizzo che fa riferimento a IDIndirizzo( IndirizzoSpedizione).

**Pagamenti**

Attributi: IDPagamento (PK), IDOrdine (FK), Importo, MetodoPagamento, DataPagamento

Chiavi esterne: IDOrdine che fa riferimento a Ordini(IDOrdine)

**Relazioni:**

**Utenti - Ordini**

Un utente può effettuare molti ordini, ma ogni ordine è effettuato da un singolo utente.

**Ordini - DettaglioOrdine**

Un ordine contiene uno o più dettagli di prodotto (prodotti ordinati), ma ogni dettaglio fa riferimento a un unico ordine.

**DettaglioOrdine - Prodotti**

Un dettaglio di ordine fa riferimento a un prodotto specifico, ma ogni prodotto può essere incluso in molti dettagli di ordine.

**Dipendenti - Ordini**

Un dipendente può gestire molti ordini, ma ogni ordine è gestito da un singolo dipendente.

**Utenti - IndirizzoSpedizione**

Un utente può avere uno o più indirizzi, e un indirizzo può appartenere a uno o più utenti quindi si procede con la creazione di una tabella di appoggio **UtenteIndirizzo**

**Utenti-Utenteindirizzo**

Ogni utente può avere zero o più righe nella tabella "UtenteIndirizzo", ognuna rappresentante un indirizzo di spedizione condiviso dall'utente.

Ogni riga nella tabella "UtenteIndirizzo" è associata a un utente e a un indirizzo di spedizione, riflettendo questa connessione.

**IndirizzoSpedizione-Utenteindrizzo**

Un indirizzo di spedizione può essere associato a uno o più utenti attraverso le righe nella tabella "UtenteIndirizzo". Ogni riga nella tabella "UtenteIndirizzo" collega un utente a un indirizzo di spedizione.

**Ordini-Pagamenti**

Un ordine può essere associato solo ad un pagamento, e un pagamento può riferirsi ad un solo ordine in quanto l’ordine viene completato nel momento in cui viene verificato il pagamento

Immagine che contiene diagramma, testo, Piano, linea

Descrizione generata automaticamente

**CREATE TABLE Utenti (**

**IDUtente INT PRIMARY KEY,**

**Nome VARCHAR(50),**

**Cognome VARCHAR(50),**

**Email VARCHAR(100),**

**DataRegistrazione DATE);**

**CREATE TABLE Prodotti (**

**IDProdotto INT PRIMARY KEY,**

**Nome VARCHAR(100),**

**Descrizione TEXT,**

**Prezzo FLOAT);**

**CREATE TABLE Dipendenti (**

**IDDipendente INT PRIMARY KEY AUTO\_INCREMENT,**

**Nome VARCHAR(50),**

**Cognome VARCHAR(50),**

**Ruolo VARCHAR(50));**

**CREATE TABLE Ordini (**

**IDOrdine INT PRIMARY KEY,**

**IDUtente INT,**

**IDDipendente INT,**

**DataOrdine DATE,**

**Totale DECIMAL(10, 2),**

**FOREIGN KEY (IDUtente) REFERENCES Utenti(IDUtente)on delete set null,**

**FOREIGN KEY (IDDipendente) REFERENCES Dipendenti(IDDipendente) on update cascade on delete set**

**Null);**

**CREATE TABLE DettaglioOrdine (**

**IDDettaglio INT PRIMARY KEY AUTO\_INCREMENT,**

**IDOrdine INT,**

**IDProdotto INT,**

**Quantità INT,**

**FOREIGN KEY (IDOrdine) REFERENCES Ordini(IDOrdine) on delete cascade,**

**FOREIGN KEY (IDProdotto) REFERENCES Prodotti(IDProdotto) on update cascade on delete restrict);**

**CREATE TABLE IndirizzoSpedizione (**

**IDIndirizzo INT PRIMARY KEY,**

**IDUtente INT,**

**Via VARCHAR(100),**

**Città VARCHAR(50),**

**CAP VARCHAR(10),**

**Paese VARCHAR(50));**

**CREATE TABLE UtenteIndirizzo (**

**IDUtenteIndirizzo INT PRIMARY KEY,**

**IDUtente INT,**

**IDIndirizzo INT,**

**FOREIGN KEY (IDUtente) REFERENCES Utenti(IDUtente**) **on update cascade on delete set null**,

**FOREIGN KEY (IDIndirizzo) REFERENCES IndirizzoSpedizione(IDIndirizzo) on update cascade on delete**

**set null);**

**CREATE TABLE Pagamenti (**

**IDPagamento INT PRIMARY KEY,**

**IDOrdine INT UNIQUE,**

**Importo DECIMAL(10, 2),**

**MetodoPagamento VARCHAR(50),**

**DataPagamento DATE,**

**FOREIGN KEY (IDOrdine) REFERENCES Ordini(IDOrdine) ON UPDATE CASCADE ON DELETE SET NULL);**

**INSERTING ROWS**

-- Inserimento di dati fittizi nella tabella Utenti

INSERT INTO Utenti (IDUtente, Nome, Cognome, Email, DataRegistrazione)

VALUES

(1, 'Alice', 'Rossi', 'alice@email.com', '2023-08-01'),

(2, 'Bob', 'Verdi', 'bob@email.com', '2023-08-02'),

(3, 'Carla', 'Bianchi', 'carla@email.com', '2023-08-03'),

(4, 'Davide', 'Gialli', 'davide@email.com', '2023-08-04'),

(5, 'Elena', 'Neri', 'elena@email.com', '2023-08-05'),

(6, 'Francesco', 'Russo', 'francesco@email.com', '2023-08-06'),

(7, 'Giulia', 'Marroni', 'giulia@email.com', '2023-08-07'),

(8, 'Luca', 'Verdi', 'luca@email.com', '2023-08-08'),

(9, 'Marta', 'Bianchi', 'marta@email.com', '2023-08-09'),

(10, 'Nicola', 'Rossi', 'nicola@email.com', '2023-08-10');

-- Inserimento di dati fittizi nella tabella Prodotti

INSERT INTO Prodotti (IDProdotto, Nome, Descrizione, Prezzo)

VALUES

(1, 'Prodotto A', 'Descrizione del Prodotto A', 19.99),

(2, 'Prodotto B', 'Descrizione del Prodotto B', 29.99),

(3, 'Prodotto C', 'Descrizione del Prodotto C', 9.99),

(4, 'Prodotto D', 'Descrizione del Prodotto D', 14.99),

(5, 'Prodotto E', 'Descrizione del Prodotto E', 24.99),

(6, 'Prodotto F', 'Descrizione del Prodotto F', 34.99),

(7, 'Prodotto G', 'Descrizione del Prodotto G', 39.99),

(8, 'Prodotto H', 'Descrizione del Prodotto H', 17.99),

(9, 'Prodotto I', 'Descrizione del Prodotto I', 12.99),

(10, 'Prodotto J', 'Descrizione del Prodotto J', 14.99);

-- Inserimento di dati fittizi nella tabella Dipendenti

INSERT INTO Dipendenti (Nome, Cognome, Ruolo)

VALUES

('Mario', 'Rossi', 'Manager'),

('Luigi', 'Verdi', 'Venditore'),

('Giovanna', 'Bianchi', 'Assistente'),

('Sara', 'Gialli', 'Venditore'),

('Antonio', 'Neri', 'Manager'),

('Laura', 'Russo', 'Venditore'),

('Alessio', 'Marroni', 'Assistente'),

('Valentina', 'Verdi', 'Manager'),

('Riccardo', 'Bianchi', 'Venditore'),

('Elisa', 'Rossi', 'Assistente');

-- Inserimento di dati fittizi nella tabella Ordini

INSERT INTO Ordini (IDOrdine, IDUtente, IDDipendente, DataOrdine, Totale)

VALUES

(1, 1, 1, '2023-08-10', 49.98),

(2, 2, 2, '2023-08-11', 59.98),

(3, 3, 1, '2023-08-12', 29.97),

(4, 4, 3, '2023-08-13', 34.96),

(5, 5, 2, '2023-08-14', 44.95),

(6, 6, 1, '2023-08-15', 64.94),

(7, 7, 3, '2023-08-16', 74.93),

(8, 8, 2, '2023-08-17', 54.92),

(9, 9, 1, '2023-08-18', 84.91),

(10, 10, 3, '2023-08-19', 24.90);

-- Inserimento di dati fittizi nella tabella DettaglioOrdine

INSERT INTO DettaglioOrdine (IDOrdine, IDProdotto, Quantità)

VALUES

(1, 1, 2),

(1, 3, 1),

(2, 2, 3),

(2, 4, 2),

(3, 1, 1),

(3, 5, 2),

(4, 2, 1),

(4, 6, 3),

(5, 3, 2),

(5, 7, 1);

-- Inserimento di dati fittizi nella tabella IndirizzoSpedizione

INSERT INTO IndirizzoSpedizione (IDIndirizzo, Via, Città, CAP, Paese)

VALUES

(1, 'Via Roma 123', 'Milano', '20123', 'Italia'),

(2, 'Via Venezia 45', 'Roma', '00100', 'Italia'),

(3, 'Via Firenze 78', 'Napoli', '80100', 'Italia'),

(4, 'Via Torino 12', 'Torino', '10100', 'Italia'),

(5, 'Via Genova 67', 'Genova', '16100', 'Italia'),

(6, 'Via Bologna 34', 'Bologna', '40100', 'Italia'),

(7, 'Via Palermo 56', 'Palermo', '90100', 'Italia'),

(8, 'Via Milano 89', 'Milano', '20123', 'Italia'),

(9, 'Via Napoli 23', 'Napoli', '80100', 'Italia'),

(10, 'Via Roma 67', 'Roma', '00100', 'Italia');

-- Inserimento di dati fittizi nella tabella UtenteIndirizzo

INSERT INTO UtenteIndirizzo (IDUtenteIndirizzo, IDUtente, IDIndirizzo)

VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 2),

(4, 4, 3),

(5, 5, 3),

(6, 6, 4),

(7, 7, 5),

(8, 8, 6),

(9, 9, 7),

(10, 10, 7);

-- Inserimento di dati fittizi nella tabella Pagamenti

INSERT INTO Pagamenti (IDPagamento, IDOrdine, Importo, MetodoPagamento, DataPagamento)

VALUES

(1, 1, 49.98, 'Carta di credito', '2023-08-10'),

(2, 2, 59.98, 'PayPal', '2023-08-11'),

(3, 3, 29.97, 'Bonifico bancario', '2023-08-12'),

(4, 4, 34.96, 'Carta di credito', '2023-08-13'),

(5, 5, 44.95, 'PayPal', '2023-08-14'),

(6, 6, 64.94, 'Bonifico bancario', '2023-08-15'),

(7, 7, 74.93, 'Carta di credito', '2023-08-16'),

(8, 8, 54.92, 'PayPal', '2023-08-17'),

(9, 9, 84.91, 'Bonifico bancario', '2023-08-18'),

(10, 10, 24.90, 'Carta di credito', '2023-08-19');

**VIEWS**

-- 1. View per identificare i dipendenti che non hanno mai gestito un ordine

**CREATE VIEW DipendentiNOOrdini AS**

**SELECT d.IDDipendente, d.Nome, d.Cognome**

**FROM Dipendenti d**

**LEFT JOIN Ordini o ON d.IDDipendente = o.IDDipendente**

**WHERE o.IDOrdine IS NULL;**

-- 2. View per visualizzare il dettaglio di un ordine con nomi dei prodotti

**CREATE VIEW DettaglioOrdineDO AS**

**SELECT**

**O.IDOrdine,**

**U.Nome AS NomeUtente,**

**U.Cognome AS CognomeUtente,**

**P.Nome AS NomeProdotto,**

**D.Quantità,**

**O.DataOrdine,**

**O.Totale**

**FROM Ordini O JOIN Utenti U ON O.IDUtente = U.IDUtente**

**JOIN**

**DettaglioOrdine D ON O.IDOrdine = D.IDOrdine**

**JOIN**

**Prodotti P ON D.IDProdotto = P.IDProdotto;**

--3. View per mostrare gli ordini più recenti

**CREATE VIEW OrdiniRecenti AS**

**SELECT IDOrdine, IDDipendente, DataOrdine, Totale**

**FROM Ordini**

**WHERE DataOrdine >= DATE\_SUB(CURDATE(), INTERVAL 2 WEEK);**

**QUERY**

-- 1. Visualizzare l'elenco dei prodotti ordinati da ciascun utente

**SELECT U.Nome, U.Cognome, P.Nome AS NomeProdotto, D.Quantità**

**FROM Utenti U**

**JOIN Ordini O ON U.IDUtente = O.IDUtente**

**JOIN DettaglioOrdine D ON O.IDOrdine = D.IDOrdine**

**JOIN Prodotti P ON D.IDProdotto = P.IDProdotto**

**ORDER BY U.Nome, U.Cognome;**

-- 2. Trovare il totale delle vendite per ciascun prodotto

**SELECT P.Nome AS NomeProdotto, SUM(D.Quantità) AS QuantitàVenduta, SUM(D.Quantità \* P.Prezzo) AS RicavoTotale**

**FROM Prodotti P**

**JOIN DettaglioOrdine D ON P.IDProdotto = D.IDProdotto**

**GROUP BY P.Nome, P.Prezzo**

**ORDER BY RicavoTotale DESC;**

-- 3. Visualizzare i dipendenti che hanno gestito almeno un ordine

**SELECT D.Nome, D.Cognome, COUNT(O.IDOrdine) AS NumeroOrdiniGestiti**

**FROM Dipendenti D**

**LEFT JOIN Ordini O ON D.IDDipendente = O.IDUtente**

**GROUP BY D.IDDipendente**

**HAVING NumeroOrdiniGestiti > 0**

**ORDER BY NumeroOrdiniGestiti DESC;**

-- 4. Trovare gli ordini effettuati nell'ultimo mese

**SELECT O.IDOrdine, U.Nome, U.Cognome, O.DataOrdine, O.Totale**

**FROM Ordini O**

**JOIN Utenti U ON O.IDUtente = U.IDUtente**

**WHERE O.DataOrdine >= DATE\_SUB(NOW(), INTERVAL 1 MONTH)**

**ORDER BY O.DataOrdine DESC;**

-- 5. Trovare il totale delle vendite per ogni mese dell'anno corrente

**SELECT MONTH(O.DataOrdine) AS Mese, SUM(O.Totale) AS RicavoMese**

**FROM Ordini O**

**WHERE YEAR(O.DataOrdine) = YEAR(NOW())**

**GROUP BY Mese**

**ORDER BY Mese;**

--6. Visualizzare l’elenco dei prodotti più venduti in ordine decrescente di quantità vendute

**SELECT P.Nome AS NomeProdotto, SUM(D.Quantità) AS QuantitàVendute**

**FROM Prodotti P**

**JOIN DettaglioOrdine D ON P.IDProdotto = D.IDProdotto**

**GROUP BY P.IDProdotto**

**ORDER BY QuantitàVendute DESC;**

--7.Visualizzare il totale delle vendite mensili dell’anno corrente

**SELECT MONTH(O.DataOrdine) AS Mese, SUM(O.Totale) AS RicavoMese**

**FROM Ordini O**

**WHERE YEAR(O.DataOrdine) = YEAR(NOW()) GROUP BY Mese ORDER BY Mese**

--8. Scovare l’utente che ha effettuato più ordini

**SELECT U.Nome, U.Cognome, COUNT(O.IDOrdine) AS NumeroOrdini**

**FROM Utenti U**

**LEFT JOIN Ordini O ON U.IDUtente = O.IDUtente**

**GROUP BY U.IDUtente**

**ORDER BY NumeroOrdini DESC**

**LIMIT 1;**

--9. Individuare i prodotti nel carrello di un utente specifico

**SELECT P.Nome AS NomeProdotto, D.Quantità AS QuantitàNelCarrello**

**FROM Utenti U**

**JOIN Ordini O ON U.IDUtente = O.IDUtente**

**JOIN DettaglioOrdine D ON O.IDOrdine = D.IDOrdine**

**JOIN Prodotti P ON D.IDProdotto = P.IDProdotto**

**WHERE U.Nome = 'NomeUtente' AND U.Cognome = 'CognomeUtente' AND O.DataOrdine IS NULL;**

--10. Visualizzare l’elenco dei dipendenti che hanno gestito almeno un ordine nel mese corrente

**SELECT D.Nome, D.Cognome, COUNT(O.IDOrdine) AS NumeroOrdiniGestiti**

**FROM Dipendenti D**

**LEFT JOIN Ordini O ON D.IDDipendente = O.IDUtente**

**WHERE MONTH(O.DataOrdine) = MONTH(NOW())**

**GROUP BY D.IDDipendente**

**HAVING NumeroOrdiniGestiti > 0**

**ORDER BY NumeroOrdiniGestiti DESC;**