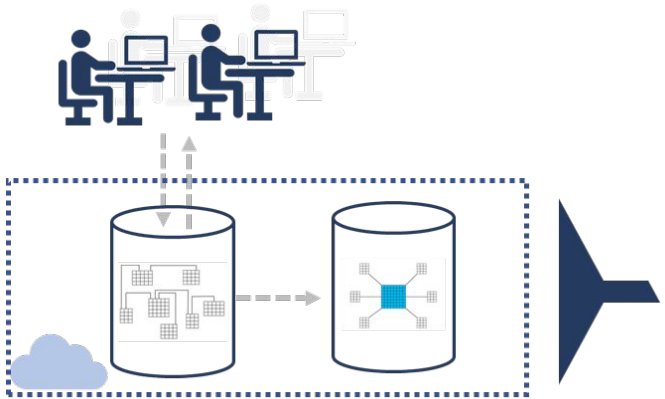
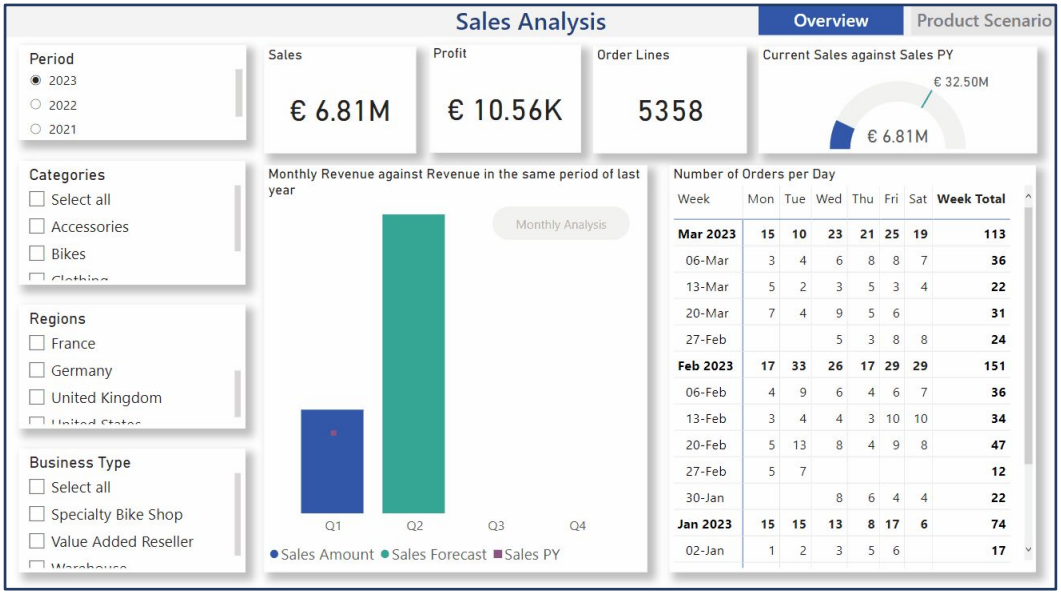
EPICODE

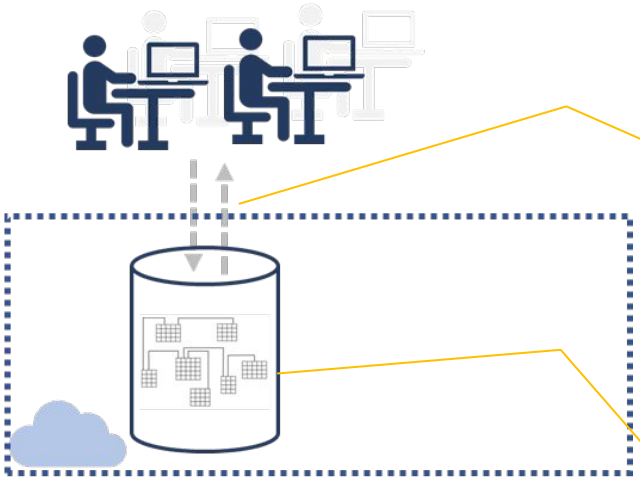
Progettazione Concettuale di un DB

Il contesto




Sales Analysis
Overview
Product Scenario
Period
2023
2022
2021
Sales
€ 6.81M
Profit
€ 10.56K
Order Lines
5358
Current Sales against Sales PY
€ 32.50M
€ 6.81M
Categories
Select all
Accessories
Bikes
Regions
France
Germany
United Kingdom
Business Type
Select all
Specialty Bike Shop
Value Added Reseller
Monthly Revenue against Revenue in the same period of last year
Monthly Analysis
Q1
Q2
Q3
Q4
Sales Amount
Sales Forecast
Sales PY
Number of Orders per Day
Week | Mon | Tue | Wed | Thu | Fri | Sat | Week Total
Mar 2023 | 15 | 10 | 23 | 21 | 25 | 19 | 113
06-Mar | 3 | 4 | 6 | 8 | 8 | 7 | 36
13-Mar | 5 | 2 | 3 | 5 | 3 | 4 | 22
20-Mar | 7 | 4 | 9 | 5 | 6 | | 31
27-Feb | | | 5 | 3 | 8 | 8 | 24
Feb 2023 | 17 | 33 | 26 | 17 | 29 | 29 | 151
06-Feb | 4 | 9 | 6 | 4 | 6 | 7 | 36
13-Feb | 3 | 4 | 4 | 3 | 10 | 10 | 34
20-Feb | 5 | 13 | 8 | 4 | 9 | 8 | 47
27-Feb | 5 | 7 | | | | | 12
30-Jan | | | 8 | 6 | 4 | 4 | 22
Jan 2023 | 15 | 15 | 13 | 8 | 17 | 6 | 74
02-Jan | 1 | 2 | 3 | 5 | 6 | | 17

Il contesto: il database e le operazione CRUD



Durante le normali attività giornaliere svolte in azienda, da parte di utenti che utilizzano software gestionali (ERP, CRM, MES), vengono generate transazioni che occorre registrare/ memorizzare.

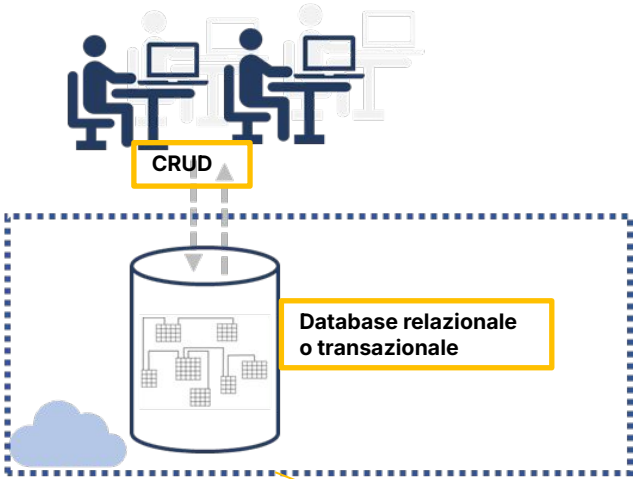
Le transazioni sono operazioni di: creazione di record (es.: registrazione di anagrafiche cliente o prodotto), lettura dati (es.: interrogazione degli ordini di vendita registrati nel mese corrente), aggiornamento dati (es.: correzione di un ordine cliente o aggiornamento di un'anagrafica cliente), eliminazione di dati (es.: eliminazione di un ordine cliente).

Queste operazioni sono dette CRUD: **C**reate, **R**ead, **U**pdate, **D**elete

I dati generati dalle operazioni CRUD sono memorizzate in **database.**

Con il termine database si intende una collezione di dati organizzata in modo logico e coerente per facilitare le operazioni di creazione/inserimento, lettura, modifica o aggiornamento, eliminazione di dati.

Il contesto: il database relazionale (transazionale) e il R(DBMS)



CRUD: operazioni di creazione/inserimento, lettura, modifica o aggiornamento, eliminazione di dati.

Database: collezione di dati organizzata in modo logico e coerente per la gestione delle operazioni CRUD.

Il tipo di database più utilizzato è il database relazionale in cui i dati sono organizzati in tabelle.

Il modello relazionale è un modello logico per la rappresentazione e l'organizzazione dei dati proposto da Edgar Codd nel 1970!

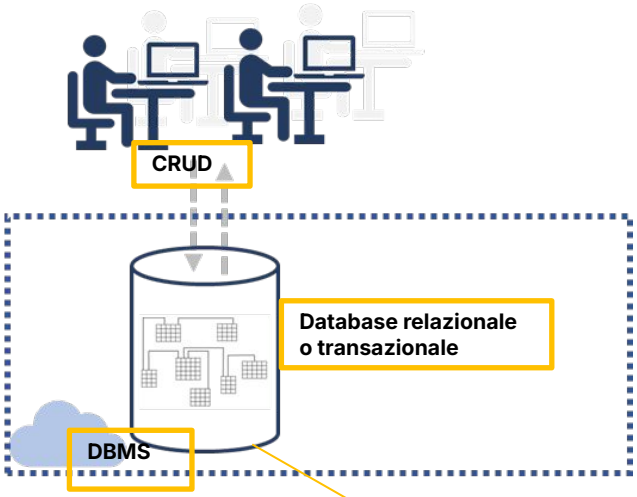
Il modello relazionale (database) è implementato su sistemi di gestione di basi dati (DBMS), detti perciò sistemi di gestione di basi dati relazionali (RDBMs).

RDBMS: **R**elational **D**atabase **M**anagement **S**ystem.

In generale, un DBMS è un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione di database.

Il sistema software è ospitato su un'architettura hardware dedicata in locale (on-premise) o cloud.

Il contesto: il sistema OLTP



Le operazioni aziendali degli utenti nell'ambito delle loro attività giornaliere (operazioni CRUD) generano transazioni.

I dati prodotti da queste transazioni sono archiviati in database relazionali (o transazionali per l'appunto).

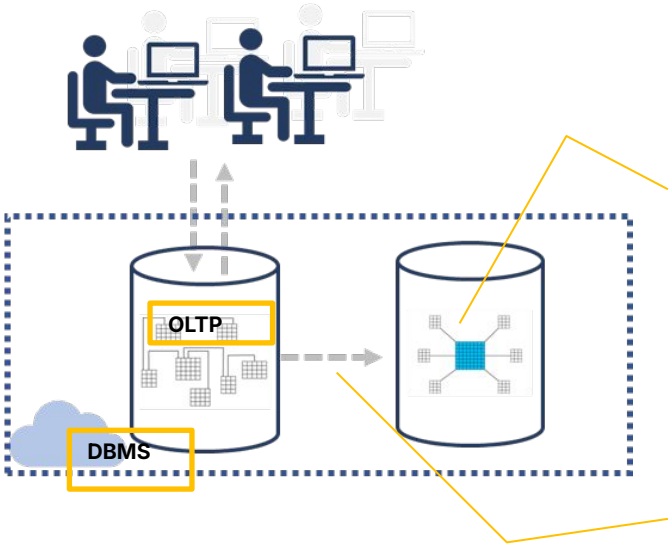
Il sistema software che consente la gestione efficiente e sicura dei database (non solo in termini di interrogazione e manipolazione dei dati ma anche in termini di utenti e accessi) è detto (R)DBMS.

La gestione dei dati transazionali tramite sistemi informatici è detta OLTP: **O**n**L**ine **T**ransactional **P**rocessing.
Lo scopo di un sistema OLTP è garantire la consistenza, l'integrità e la sicurezza delle transazioni stesse.

Sistemi OLTP sono tipici di scenari con intensa attività di scrittura quindi transazioni/operazioni CRUD brevi ma frequenti. I tempi di risposta sono rapidi (per moli di dati ridotte).

Il sistema OLTP è ottimizzato per la gestione dei dati (data entry) e non per l'analisi dei dati (reporting).

Il contesto: il sistema OLAP



Il data warehouse restituisce una visione integrata e consistente dei dati provenienti dalle diverse aree o sistemi aziendali.

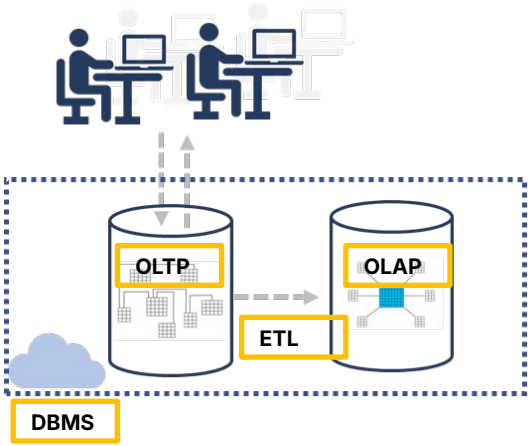
Ha una struttura multidimensionale che permette di analizzare le informazioni a diversi livelli di aggregazione e secondo diverse prospettive di analisi.

Il data warehouse è una riorganizzazione dei dati esistenti ottimizzata per l'analisi dati e quindi per la costruzione di soluzioni di business intelligence (BI).

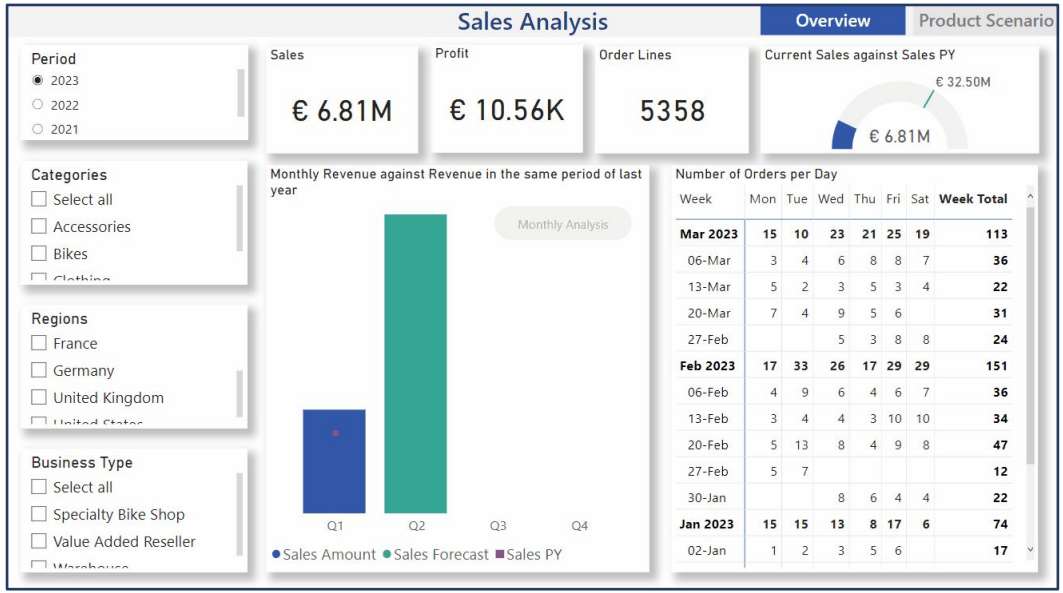
L'approccio (metodologico e software) per l'analisi complessa di grandi volumi dati a supporto della BI (e che costituisce il cuore del data warehouse) è detto OLAP: **O**n**L**ine **A**nalytical **P**rocessing.

Il data warehouse è popolato con procedure di **ETL** (Extract, Transform, Load).

Il contesto: business intelligence



ETL

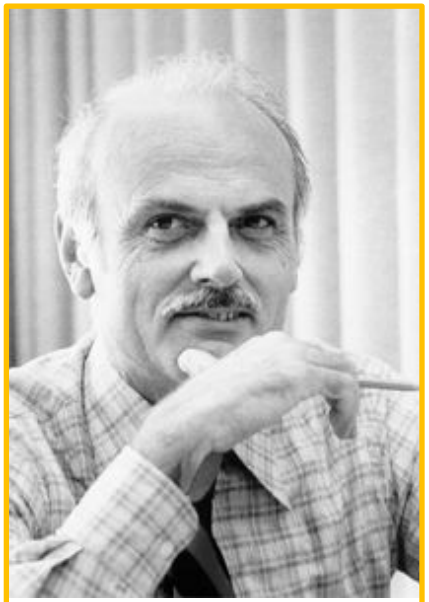


Data Analyst: acquisisce dati da sorgenti eterogenee come database (OLTP e/o OLAP), file flat (csv, txt), cartelle di lavoro Excel, dal web, ... li trasforma in informazioni (un dato diventa un'informazione quanto suggerisce un'azione) e poi in conoscenza a supporto di un processo decisionale!

Il modello relazionale

In un database relazionale i dati sono organizzati in tabelle:

- Una **tabella** o relazione è una rappresentazione formale di un concetto/evento reale tangibile o intangibile (cliente, prodotto, vendite, fornitore, ...).
- Le informazioni relative al concetto/all'evento rappresentato sono memorizzate come insieme di colonne o campi o attributi. Ogni colonna nella tabella è progettata per contenere un tipo di informazioni specifico per cui occorre definirne il formato (data type): data, stringa, intero.
- Una singola occorrenza (o manifestazione) dell'evento che la tabella descrive è detta riga o record o tupla.



Edgar Frank Codd

Il modello relazionale: tabelle e relazioni

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546
71992	15/07/2008	29546

SalesOrderDetail

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

Product

ProductID	Name	ListPrice
706	HL Road Frame	1431,5
715	Jersey, L	49,99
836	ML Road Frame	594,83
875	Racing Socks, L	8,99
881	Classic Jersey, S	53,99

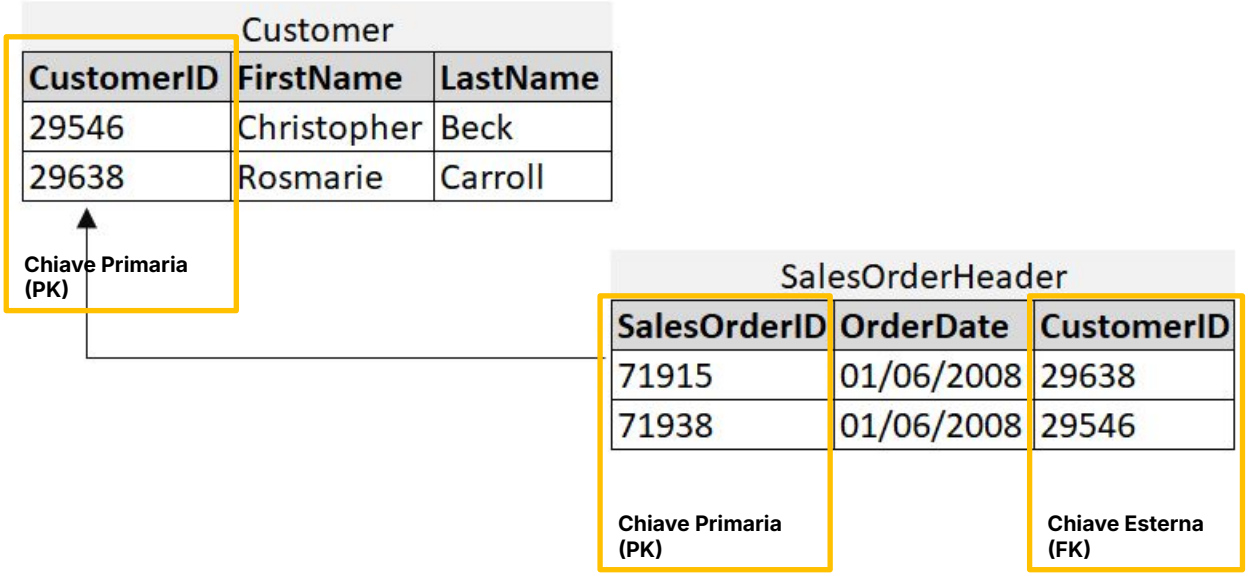
Le tabelle rappresentano oggetti/eventi o entità del mondo reale (tipicamente eventi inerenti ad un processo aziendale).

Questi eventi non sono manifestazioni isolate ma hanno tra loro relazioni.

Le relazioni tra tabelle descrivono i processi reali che il database deve modellare.

Ma come si implementano le relazioni?

Il modello relazionale: tabelle e relazioni



L'implementazione di una relazione avviene individuando una chiave primaria e una chiave esterna.

La chiave primaria (PK) di una tabella è un attributo (o un insieme di attributi) della stessa che permette di identificare in maniera univoca ogni record della tabella.

I dati devono essere accessibili senza ambiguità. I valori di una PK sono univoci (non ammette valori ridondanti).

La chiave esterna di una tabella è un attributo (o un insieme di attributi) che fa riferimento a una chiave di un'altra tabella.

Il modello relazionale: integrità referenziale

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

Chiave Primaria (PK)

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546

Chiave Esterna (FK)

L'integrità referenziale dei dati, elimina, quando si hanno relazioni associate tra loro attraverso FK, errori di inserimento, cancellazione o modifica di dati collegati tra loro.

Definire una FK (riferita ad una PK) consente di esplicitare relazioni uno a molti tra tabelle e garantire l'integrità referenziale dei dati.

Il campo **CustomerID** della tabella **Customer** è **PK** (non è possibile inserire un'anagrafica duplicata altrimenti si avrebbero informazioni ridondanti, ambigue e non consistenti e non sarebbe possibile distinguere i record in modo univoco).

Il campo **CustomerID** della tabella **SalesOrderHeader** è **FK** (un ordine cliente deve essere associato ad un codice cliente esistente in anagrafica altrimenti si avrebbero dati errati, incompleti, non affidabili)

Il modello relazionale: integrità referenziale

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

Chiave Primaria (PK)

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546

Chiave Esterna (FK)

La tabella **SalesOrderHeader** è referenziata dalla tabella **Customer** (referenziante).

In altre parole, dato il vincolo di FK i record tabella **SalesOrderHeader** sono 'figli' della tabella '**Customer**' (non ci possono essere codici cliente in **SalesOrderHeader** che non sono stati definiti in **Customer**).

La relazione è tra le due tabelle è uno a molti: ad un'anagrafica cliente sono associate una o molte transazioni di vendita (anche nessuna!)

Il modello relazionale: esempio di chiave composita

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546
71992	15/07/2008	29546

SalesOrderDetail

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

Chiave Primaria (PK)

Il campo SalesOrderID in SalesOrderHeader è PK di SalesOrderHeader.

L'insieme degli attributi SalesOrderID e SalesOrderDetailID è PK di SalesOrderDetail. Se osserviamo i valori di ciascun campo notiamo che non sono univoci. Il singolo campo non puo' essere una PK perché contiene ridondanze... la combinazione è univoca!

Il campo SalesOrderID di SalesOrderDetail è la FK alla tabella SalesOrderHeader.

Il modello relazionale: integrità referenziale

La PK è un vincolo che garantisce che non vengano inseriti valori duplicati o null. In altre parole, vieta l'inserimento di un valore se questo esiste già!

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

La FK elimina errori di inserimento, modifica o cancellazione.

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546

Il modello relazionale: schema OLTP

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

SalesOrderHeader

SalesOrderID	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546
71992	15/07/2008	29546

SalesOrderDetail

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

Product

ProductID	Name	ListPrice
706	HL Road Frame	1431,5
715	Jersey, L	49,99
836	ML Road Frame	594,83
875	Racing Socks, L	8,99
881	Classic Jersey, S	53,99

L'obiettivo del modello relazionale è ottenere una rappresentazione dei dati completa e consistente, cioè i dati devono essere significativi, utilizzabili.

Un altro aspetto caratterizzante i modelli relazionali (transazionali) è l'assenza o la minimizzazione di dati ridondanti.

Il modello relazionale: schema OLTP

ProductID	Name	ListPrice	Category
706	HL Road Frame	1431,5	Road Frames
715	Jersey, L	49,99	Jerseys
836	ML Road Frame	594,83	Road Frames
875	Racing Socks, L	8,99	Socks
881	Classic Jersey, S	53,99	erseys

CategoryID	Name
18	Road Frames
25	Jerseys
27	Socks

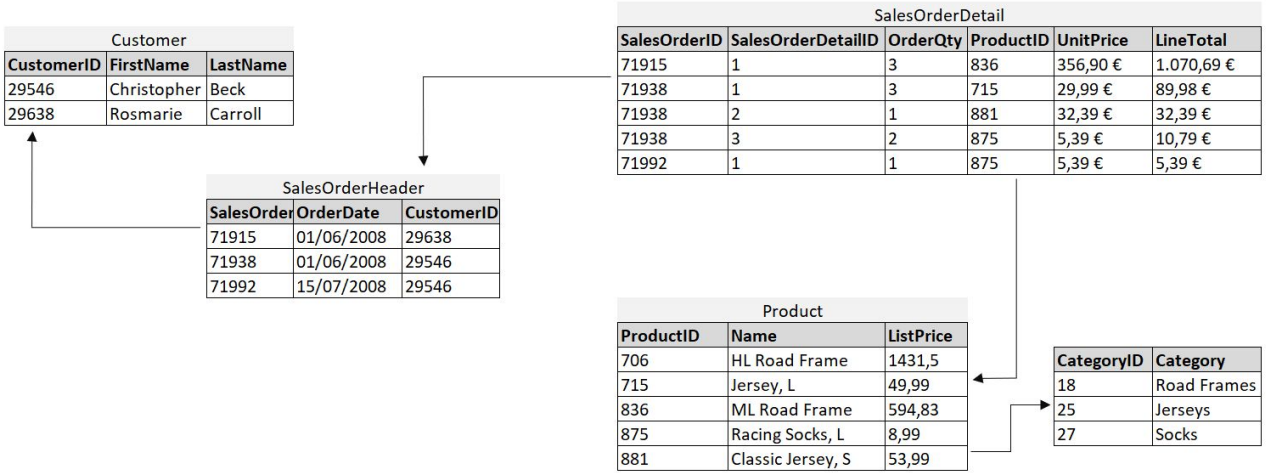
ProductID	Name	ListPrice	CategoryID
706	HL Road Frame	1431,5	18
715	Jersey, L	49,99	25
836	ML Road Frame	594,83	18
875	Racing Socks, L	8,99	27
881	Classic Jersey, S	53,99	25

La ridondanza dei dati può causare incoerenze nei dati perchè lo stesso 'pezzo di informazione' potrebbe essere memorizzato in modo diverso in più posizioni anziché essere un qualcosa di univoco preciso e affidabile.

Il processo di minimizzazione della ridondanza dati nei database transazionali (OLTP) è chiamato normalizzazione.

Concettualmente significa distribuire i dati della stessa entità in più tabelle.

Il modello relazionale: schema OLTP



Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

SalesOrderHeader

SalesOrder	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546
71992	15/07/2008	29546

SalesOrderDetail

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

Product

ProductID	Name	ListPrice
706	HL Road Frame	1431,5
715	Jersey, L	49,99
836	ML Road Frame	594,83
875	Racing Socks, L	8,99
881	Classic Jersey, S	53,99

CategoryID	Category
18	Road Frames
25	Jerseys
27	Socks

Un database relazionale transazionale (OLTP) è estremamente normalizzato, in tal modo:

- Minimizza la ridondanza dei dati
- Garantisce l'integrità referenziale dei dati

Una base dati progettata in questo modo, di conseguenza, ottimizza la gestione del dato e la corretta esecuzione delle operazioni CRUD.

Il modello relazionale: schema OLTP

Customer

CustomerID	FirstName	LastName
29546	Christopher	Beck
29638	Rosmarie	Carroll

SalesOrderHeader

SalesOrder	OrderDate	CustomerID
71915	01/06/2008	29638
71938	01/06/2008	29546
71992	15/07/2008	29546

SalesOrderDetail

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

Product

ProductID	Name	ListPrice
706	HL Road Frame	1431,5
715	Jersey, L	49,99
836	ML Road Frame	594,83
875	Racing Socks, L	8,99
881	Classic Jersey, S	53,99

CategoryID	Category
18	Road Frames
25	Jerseys
27	Socks

Ma come si progetta una base dati come questa??

Il modello relazionale: progettazione di un modello relazionale transazionale

Ma come si progetta una base dati come questa??

La modellazione di una base prevede tipicamente 3 step:

1. Modellazione concettuale
2. Modellazione logica
3. Implementazione fisica

Modellazione concettuale: E/R

Modellazione concettuale

- Il modello concettuale descrive la realtà di interesse che la base dati deve modellare.
- Il modello concettuale prescinde dal software (SQL Server, MySql, Oracle) e dall'architettura hardware.
- L'obiettivo della modellazione concettuale è tradurre la realtà di interesse e le esigenze aziendali.
- In questa fase bisogna concentrarsi sull'identificazione delle entità che faranno parte del database e delle relazioni esistenti tra queste entità; questa fase è fondamentale per comprendere e definire i processi aziendali.
- È una rappresentazione della struttura generale dei dati per soddisfare certi processi aziendali.
- Come si fa? La tecnica principe per la rappresentazione grafica del modello concettuale è lo schema E/R

Il modello Entity-Relationship è un modello teorico per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione https://it.wikipedia.org/wiki/Modello_E-R.

E/R: Entity

Il modello concettuale descrive la realtà di interesse che la base dati deve modellare:

- Tramite l'individuazione di entità (con i suoi attributi) e relazioni
- Indipendentemente dal DBMS

Entità

- Rappresentano classi/insiemi di oggetti del mondo reale fisici/tangibili (persone, oggetti, luoghi) oppure astratti (eventi).
- Hanno un nome che le identificano univocamente

Esempi: Cliente, Ordine di vendita, Agente, Area di vendita

E/R: Entity

Il modello concettuale descrive la realtà di interesse che la base dati deve modellare:

- Tramite l'individuazione di entità (con i suoi attributi) e relazioni
- Indipendentemente dal DBMS

Attributi

- Descrivono entità
- Ciascun attributo ha un formato (numero, stringa, data, data/ora) e una dimensione (quantità massima di cifre o caratteri ammessi). In altre parole, ciascun attributo ha un dominio che rappresenta l'insieme dei valori assunti dall'attributo.

Esempio: FirstName è un attributo dell'entità Customer il cui dominio è l'insieme delle stringhe di lunghezza 25 caratteri

- Un attributo può essere
  - Semplice
  - Multiplo
  - Composto
  - Opzionale
  - Derivato o calcolato

- Un attributo o un'insieme di attributi svolge il ruolo di identificatore (chiave primaria)

E/R: Entity

Abbiamo detto che Il modello concettuale descrive la realtà di interesse che la base dati deve modellare.

1. La prima cosa che occorre fare è dunque individuare le entità coinvolte in una certa realtà di interesse (processo aziendale). Individuare le entità significa anche definirne gli attributi che le descrivono.
2. La seconda cosa che occorre fare è individuare le relazioni tra queste entità.

La progettazione deve essere tale da garantire la minimizzazione della ridondanza dei dati e l'integrità referenziale dei dati.

Le entità devono pertanto essere normalizzate!

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.

1. **Quali sono le entità coinvolte (e come possiamo rappresentarle graficamente)?**

Vediamo degli esempi delle possibili entità coinvolte e di come si rappresentano in uno schema E/R.

E/R: Entity

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.

1. **Quali sono le entità coinvolte (e come possiamo rappresentarle graficamente)?**

Entità

- Customer: una persona oppure un'azienda-cliente che acquista prodotti e/o servizi
- Salesperson: una persona per mezzo della quale l'azienda vende prodotti e/o servizi a clienti
- Product: un item/articolo e/o servizio che l'azienda vende al cliente
- SalesOrderHeader: richiesta di acquisto da parte di un cliente per uno o più articoli e/o servizi
- SalesOrderDetail: la singola transazione di vendita ognuna delle quale indica il prodotto o articolo o item ordinato dal cliente, la quantità, il prezzo, ...

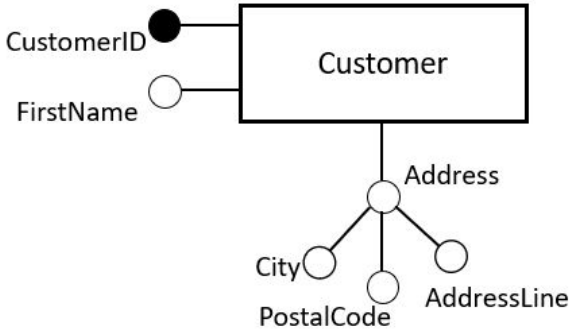
Attributi

- Customer: CustomerID, FirstName, LastName, ...
- Salesperson: SalespersonID, SalespersonName, ...
- Product: ProductID, Name, ListPrice, ...
- SalesOrderHeader: SalesOrderID, OrderDate, SalespersonID, CustomerID, ...
- SalesOrderDetail: SalesOrderID, SalesOrderDetailID, ProductID, Quantity, ...

E/R: costrutti grafici

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.



Entità: Customer

Identificatore: CustomerID

Attributo semplice: FirstName

Attributo composto: Address

Attributo composto: bisogna esporre gli attributi atomici (inscindibili che concorrono alla definizione dell'attributo composto)

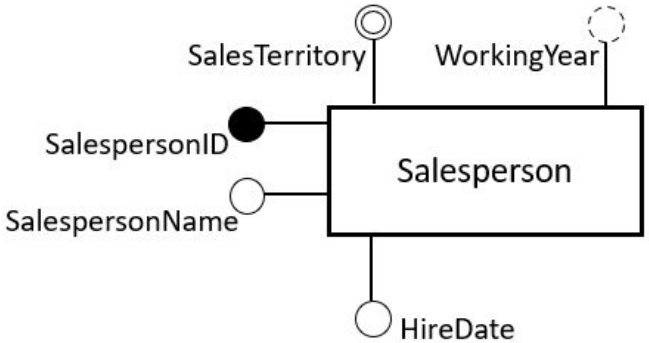
Proviamo a visualizzarne i dati:

CustomerID	FirstName	City	PostalCode	AddressLine
29546	Christopher	Toronto	M4B 1V7	52560 Free Street
29638	Rosmarie	London	SW6 SBY	586 Fulham Road

E/R: costrutti grafici

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.



Entità: Salesperson
Identificatore: SalespersonID
Attributo semplice: SalespersonName, HireDate
Attributo derivato o calcolato: WorkingYear
Attributo multiplo: SalesTerritory (differenza tra data corrente e data assunzione).

Attributo multiplo: un attributo multiplo espone molti valori per uno stesso oggetto.
Un'agente è assegnato a più regioni di vendita e una stessa regione di vendita puo' essere di competenza di più agenti.
La gestione di un attributo puo' essere molto complessa e la approfondiremo nel seguito!

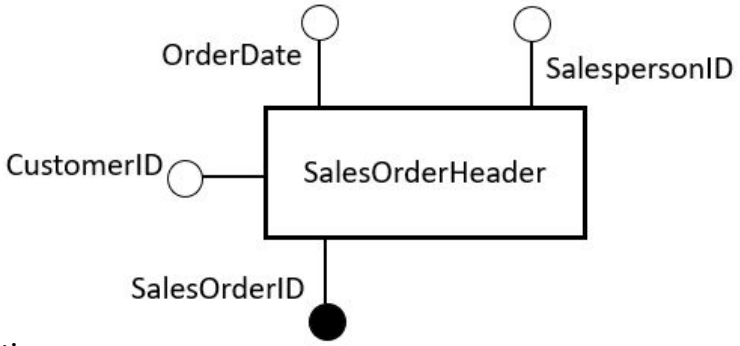
Proviamo a visualizzarne i dati:

SalespersonID	SalespersonName	HireDate	SalesTerritory	WorkingYear
290	Amy Alberts	01/01/2014	France	10
290	Amy Alberts	01/01/2014	Germany	10
290	Amy Alberts	01/01/2014	United Kingdom	10
291	Jae Pak	01/06/2017	Canada	7
291	Jae Pak	01/06/2017	France	7
291	Jae Pak	01/06/2017	United Kingdom	7

E/R: costrutti grafici

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.



Entità: SalesOrderHeader

Identificatore: SalesOrderID

Attributi: CustomerID, OrderDate, SalespersonID

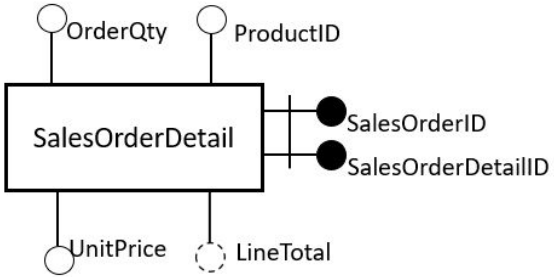
Proviamo a visualizzarne i dati:

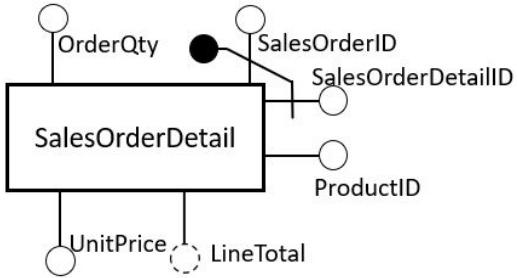
SalesOrderID	OrderDate	SalespersonID	CustomerID
71915	01/06/2008	291	29638
71938	01/06/2008	290	29546
71992	15/07/2008	290	29546

E/R: costrutti grafici

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.





Entità: SalesOrderDetail

Identificatore composito: SalesOrderID, SalesOrderDetailID

Attributo calcolato o derivato: LineTotal. L'attributo LineTotal verrà popolato in funzione di un calcolo, in questo caso il prodotto tra UnitPrice e OrderQty.

Duplice costrutto grafico per indicare una chiave composita!

Proviamo a visualizzarne i dati:

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	LineTotal
71915	1	3	836	356,90 €	1.070,69 €
71938	1	3	715	29,99 €	89,98 €
71938	2	1	881	32,39 €	32,39 €
71938	3	2	875	5,39 €	10,79 €
71992	1	1	875	5,39 €	5,39 €

E/R: Relationships

Abbiamo detto che Il modello concettuale descrive la realtà di interesse che la base dati deve modellare.

1. La prima cosa che occorre fare è dunque individuare le entità coinvolte in una certa realtà di interesse (processo aziendale). Individuare le entità significa anche definirne gli attributi che le descrivono.
2. La seconda cosa che occorre fare è individuare le relazioni tra queste entità.

La progettazione deve essere tale da garantire la minimizzazione della ridondanza dei dati e l'integrità referenziale dei dati.

Le entità devono pertanto essere normalizzate!

Supponiamo di dover creare la base dati per modellare un processo di vendita.

Il processo di vendita si riferisce a transazioni di vendita di prodotti a clienti per mezzo di agenti di vendita.

1. Quali sono le entità coinvolte (e come possiamo rappresentarle graficamente)?
2. **Quali relazioni sussistono tra le entità? Come interagiscono?**

E/R: Relationships

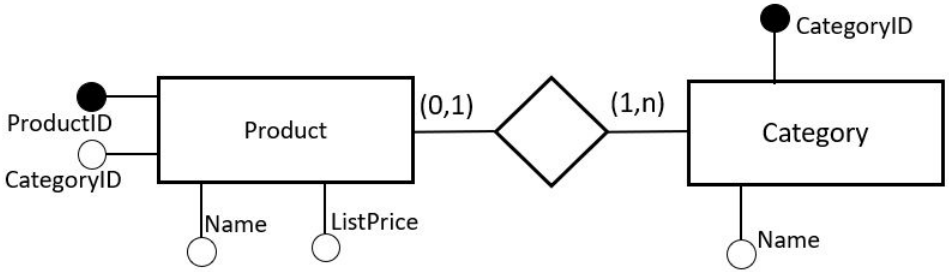
Una relazione rappresenta il legame logico tra due o più entità le entità non sono manifestazioni isolate ma 'interagiscono' tra loro.

Per ogni entità partecipante ad una relazione viene specificata una **cardinalità di relazione.** La cardinalità indica il numero minimo e massimo di 'volte' che una stessa istanza o occorrenza di un'entità partecipa alla relazione.

In altre parole, la cardinalità di una relazione indica il numero di righe in una tabella che puo' essere correlato al numero di righe in un'altra tabella.

ProductID	Name	ListPrice	CategoryID
706	HL Road Frame	1431,5	18
715	Jersey, L	49,99	25
836	ML Road Frame	594,83	18
875	Racing Socks, L	8,99	27
877	Blade	null	null
881	Classic Jersey, S	53,99	25

CategoryID	Name
18	Road Frames
25	Jerseys
27	Socks

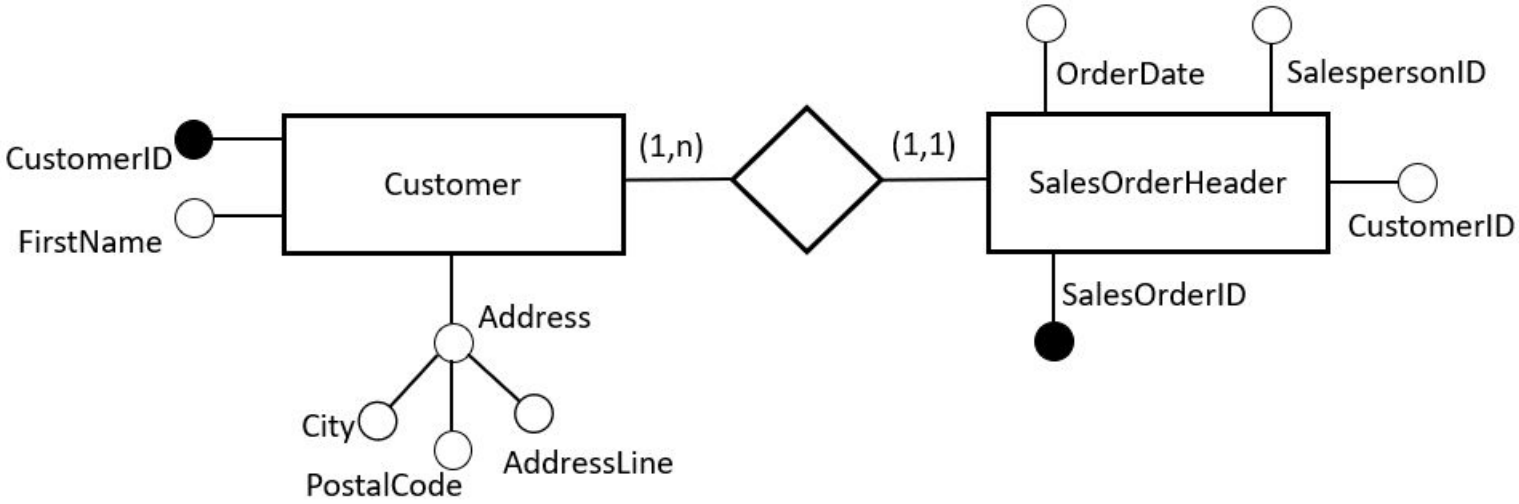


La cardinalità di partecipazione della relazione tra Product e Category minima è 0: ad un prodotto non è necessariamente associata una categoria. La cardinalità di partecipazione massima è 1: ad un prodotto è associata un'unica categoria (nella realtà potrebbe verificarsi il caso in cui uno stesso codice articolo abbia classificazioni diverse..)

La cardinalità di partecipazione della relazione tra Category e Product è 1: ad una categoria è associato almeno un prodotto. La cardinalità di partecipazione massima è *n* : ad una stessa categoria sono associati più prodotti (codici articolo diversi sono classificati allo stesso modo)

E/R

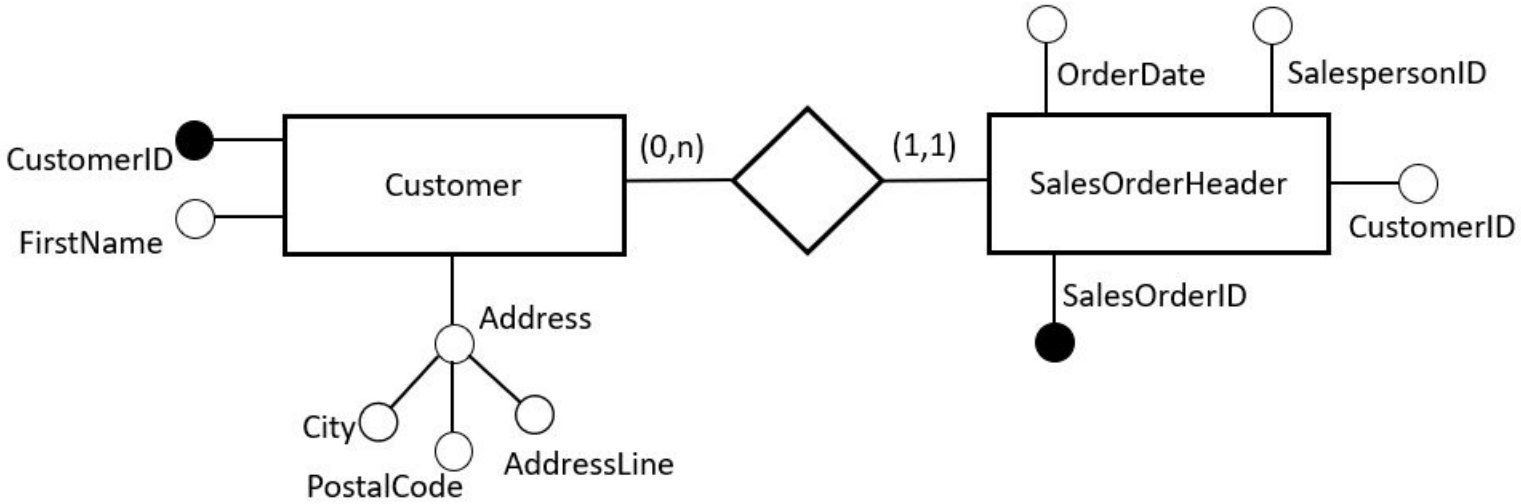
Un cliente (Customer) piazza uno o più ordini (SalesOrderHeader): one-to-many
Un ordine (SalesOrderHeader) si riferisce ad uno solo cliente (Customer): one-to-one



Partecipazione obbligatoria: il cliente in quanto tale esiste dal momento in cui effettua un ordine.

E/R

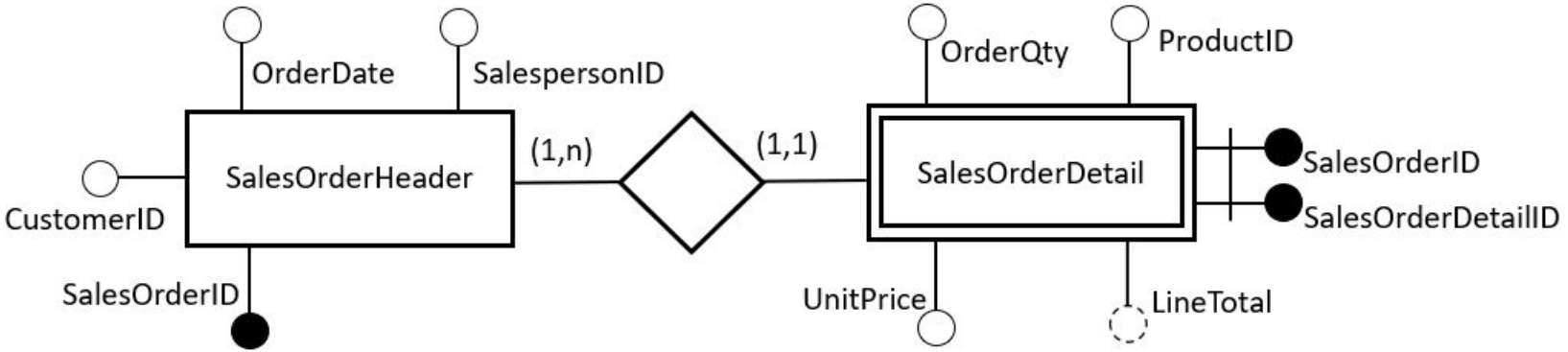
Un cliente (Customer) può piazzare uno o più ordini (SalesOrderHeader): one-to-many
Un ordine (SalesOrderHeader) si riferisce ad uno solo cliente (Customer): one-to-one



Partecipazione opzionale: la cardinalità minima è 0. Si registra in un cliente in anagrafica ma non ha piazzato ordini.

E/R

Un ordine (SalesOrderHeader) può contenere più righe di corpo, più dettagli, più transazioni (SalesOrderDetail): one-to-many
Una singola transazione (SalesOrderDetail) si riferisce ad un solo ordine (SalesOrderHeader): one-to-one

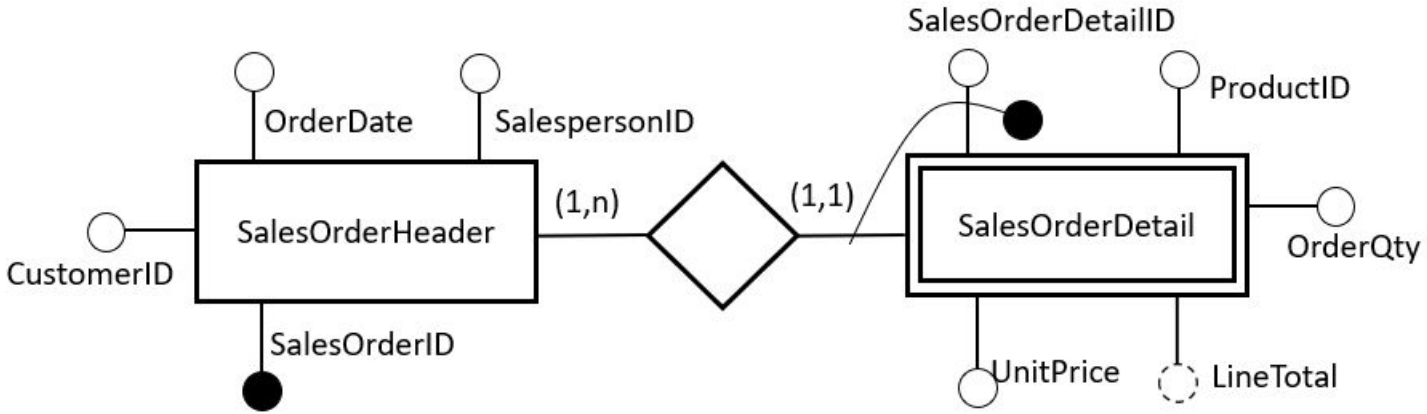


SalesOrderDetail è un'entità debole: una transazione ha luogo contestualmente alla registrazione/inserimento di un ordine.

Un'entità è detta debole se la sua esistenza dipende da un'altra entità. Un'entità debole non può essere definita dai suoi soli attributi. La PK di un'entità debole è data dalla composizione di uno o più dei suoi attributi e da una chiave esterna (della tabella da cui dipende).

E/R

Un ordine (SalesOrderHeader) può contenere più righe di corpo, più dettagli, più transazioni (SalesOrderDetail): one-to-many
Una singola transazione (SalesOrderDetail) si riferisce ad un solo ordine (SalesOrderHeader): one-to-one

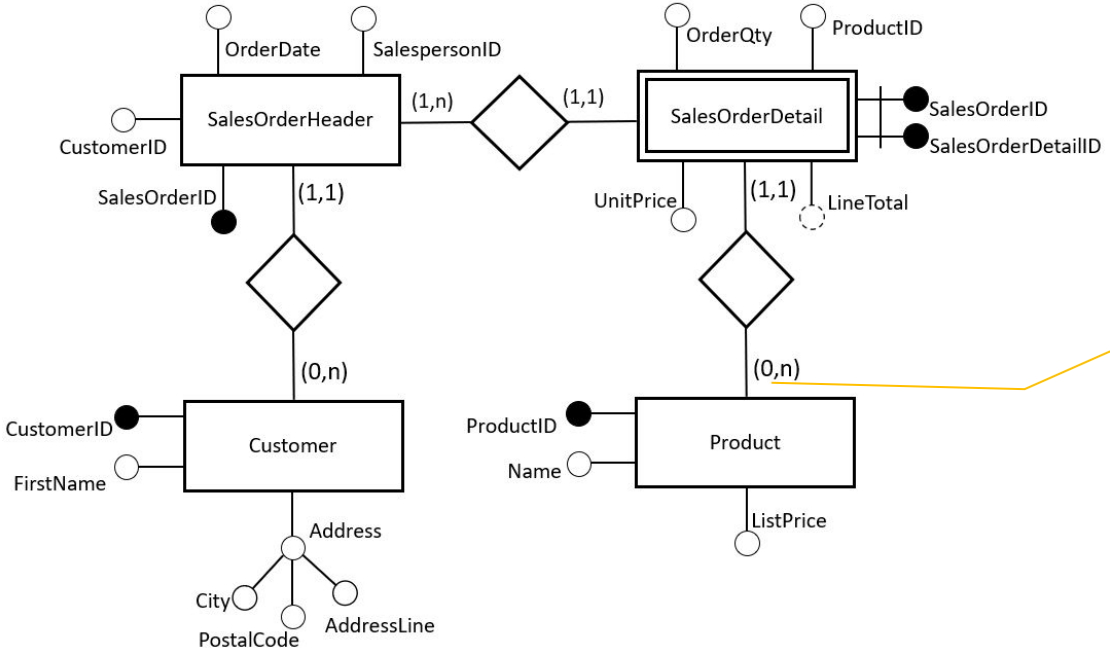


Costrutto grafico alternativo al precedente!

Un'entità è detta debole se la sua esistenza dipende da un'altra entità. Un'entità debole non può essere definita dai suoi soli attributi. La PK di un'entità debole è data dalla composizione di uno o più dei suoi attributi e da una chiave esterna (della tabella da cui dipende).

E/R

Un prodotto (Product) può essere venduto tante volte (o nessuna) per cui è contenuto in più transazioni (SalesOrderHeader): one-to-many
Una singola transazione (SalesOrderDetail) può riferirsi ad un singolo prodotto e basta (Product): one-to-one



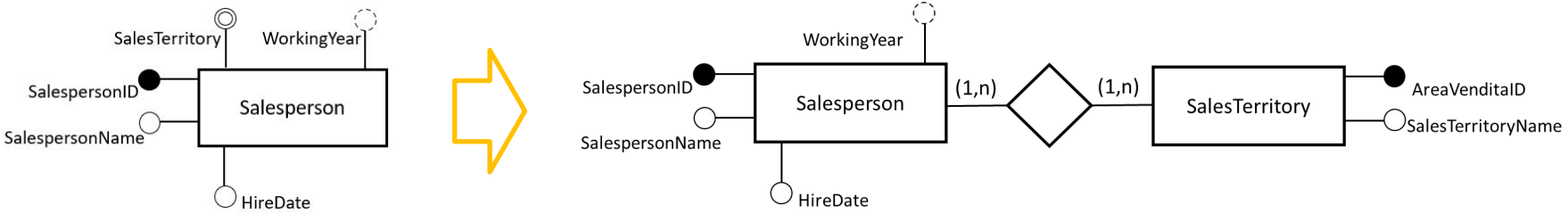
Un prodotto puo' essere venduto 0, 1 o *n* volte.

Partecipazione minima opzionale: articolo presente in anagrafica ma non vendibile.

L'entità potrebbe essere arricchita di una attributo tipo prodotto (materia prima, semilavorato o prodotto finito)

E/R

La rappresentazione concettuale più elementare per l'entità Salesperson puo' essere migliorata esplicitando tutte le entità coinvolte e le relazioni tra le stesse.



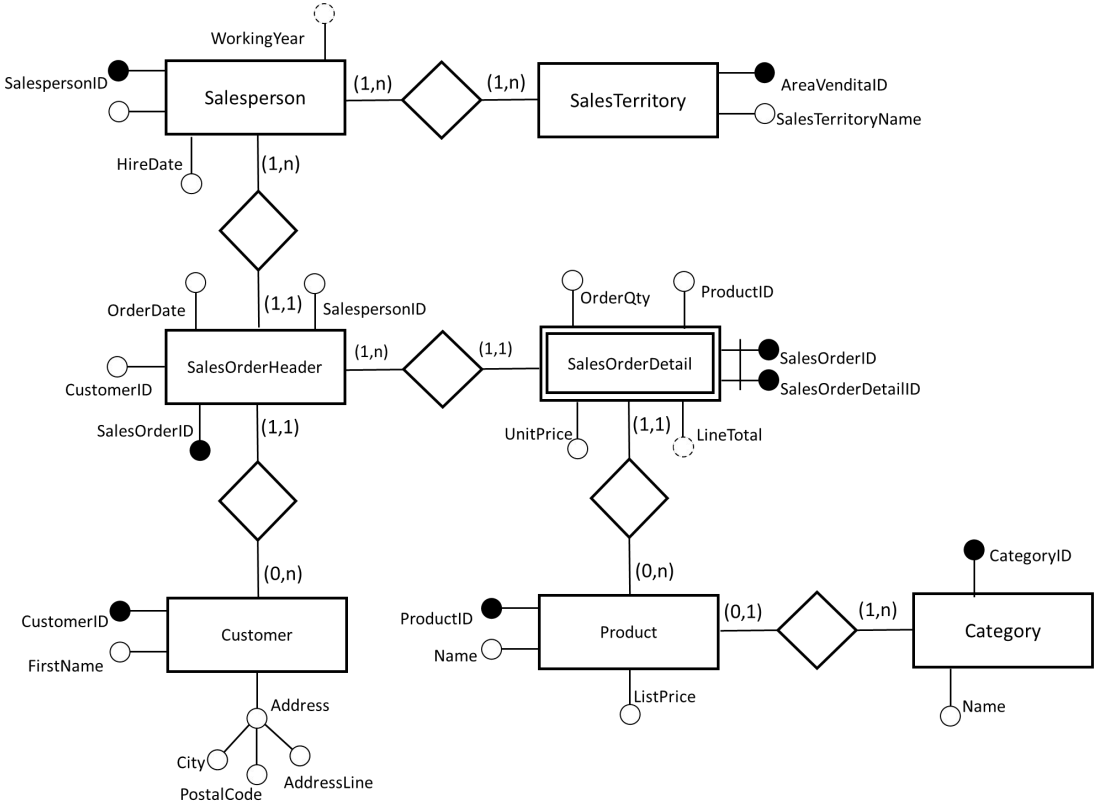
Questo è un esempio di relazione many-to-many (molti a molti):

- Ad un salesperson (agente) sono associate molte salesterritory (aree di vendita)
- Una salesterritory (un'area di vendita) è di competenza di salesperson (agenti) diversi

In un database relazionale non possono essere implementate relazioni molti a molti.

In fase di progettazione logica è necessario introdurre una tabella (di correlazione o bridge) per 'spezzare la relazione molti a molti' in relazioni uno a molti (o molti a uno)! Infatti, ogni relazione molti a molti corrisponde ad una tabella!!

E/R: recap!


WorkingYear
SalespersonID
Salesperson
(1,n)
(1,n)
SalesTerritory
AreaVenditaID
SalesTerritoryName
HireDate
(1,n)
OrderDate
(1,1)
SalespersonID
OrderQty
ProductID
CustomerID
SalesOrderHeader
(1,n)
(1,1)
SalesOrderDetail
SalesOrderID
SalesOrderDetailID
SalesOrderID
(1,1)
UnitPrice
(1,1)
LineTotal
(0,n)
(0,n)
CategoryID
CustomerID
Customer
ProductID
Name
Product
(0,1)
(1,n)
Category
FirstName
Address
ListPrice
Name
City
PostalCode
AddressLine

EPICODE

GRAZIE

Epicode