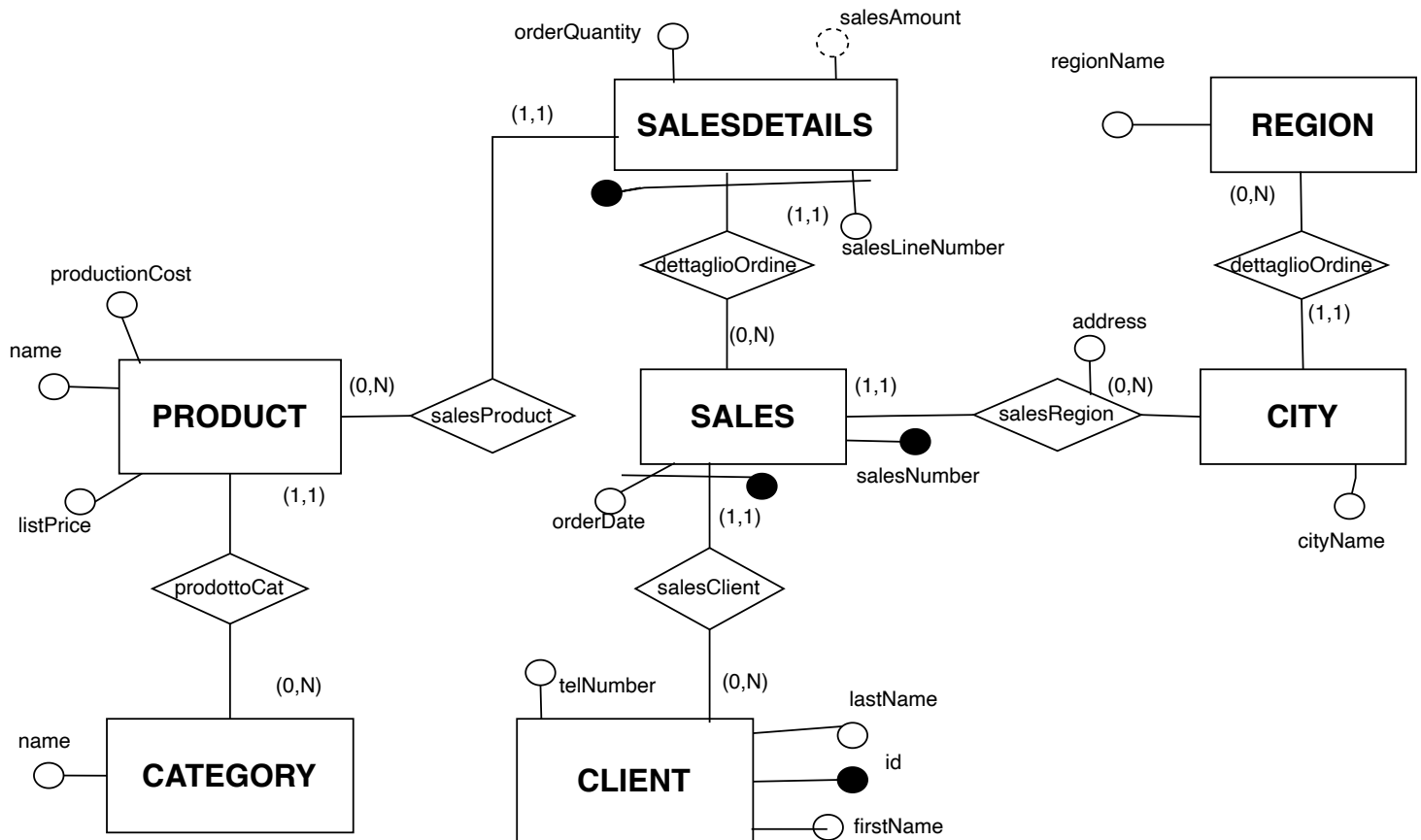


# TOYS GROUP

## ENTITY RELATIONSHIP DIAGRAM



## DESCRIZIONE ER

### ENTITA' FORTI

#### CLIENT

Client rappresenta il cliente che effettua l'ordine.

Un' istanza di Client può effettuare diversi ordini:

cardinalità (0,N) dal suo lato della relationship con Sales

#### CATEGORY

Category rappresenta la cateogria associata ad un Product.

Un' istanza di Categoria può essere associata a diversi ordini:

cardinalità (0,N) dal suo lato della relationship con Product

#### REGION

Region rappresenta la regione/lo Stato di una città.

Un' istanza di Region può essere associata a diverse città:

cardinalità (0,N) dal suo lato della relationship con City

## ENTITA' DEBOLI

### PRODUCT

Product rappresenta il prodotto che è stato ordinato.

Un'istanza di Product può essere associata a diverse SalesDetails:  
cardinalità (0,N) dal suo lato della relationship con SalesDetails

Un'istanza di Product può avere una sola Category:  
cardinalità (1,1) dal suo lato della relationship con Category

### CITY

City rappresenta la città in cui l'ordine deve essere spedito.

Un'istanza di City può essere associata a diverse Sales:  
cardinalità (0,N) dal suo lato della relationship con Sales

Un'istanza di Product può avere una sola Region:  
cardinalità (1,1) dal suo lato della relationship con Region

### SALES

Sales rappresenta l'ordine effettuato.

Un'istanza di Sales può essere associata a diverse SalesDetails:  
cardinalità (0,N) dal suo lato della relationship con SalesDetails

Un'istanza di Sales può avere un solo Client:  
cardinalità (1,1) dal suo lato della relationship con Client

Un'istanza di Sales può avere una sola City:  
cardinalità (1,1) dal suo lato della relationship con City

### SALESDetails

SalesDetails rappresenta il dettaglio di un ordine.

Un'istanza di SalesDetails può essere associata a una sola Sales:  
cardinalità (1,1) dal suo lato della relationship con Sales

Un'istanza di SalesDetails si riferisce a un solo Product:  
cardinalità (1,1) dal suo lato della relationship con Product

## VINCOLI

Non esistono due istanze uguali di SalesDetails con stesso salesNumber (dalla relationship con SalesNumber) e stesso salesLineNumber.

Non esistono due istanze uguali di Sales con stesso Client (dalla relationship con Client) e stessa orderDate.

Dalla modellazione, si è omessa la relazione possibile tra Client e City:

ovvero che un Client debba risiedere nella City in cui far spedire l'ordine.

Non inserendo la relazione, si permette che un certo Client risieda anche in una City non presente nel database e che effettui un ordine per un'altra persona, anch'essa non registrata nel db.

In questo modo, si è reso il sistema più flessibile da questo punto di vista.

Nel caso in cui fosse stata inserita la relazione si sarebbe cercato di inserire nell'implementazione fisica del db, funzioni o trigger che potessero gestire e controllare questo problema.

Non avendolo visto durante il corso, è stata presa la decisione di non implementare la relazione.

## SCHEMA LOGICO

### ENTITA' FORTI

#### • CLIENT:

- id int
- firstName varchar(40)
- lastName varchar(40)
- telNumber varchar(25)
- PK(id)

#### • CATEGORY:

- id int
- name varchar(40)
- PK(id)

#### • REGION:

- id int
- regionName varchar(40)
- PK(id)

### ENTITA' DEBOLI

#### • CITY:

- id int
- name varchar(40)
- idRegion
- PK (id)
- FK (idRegion) ref. Region(id)

#### • PRODUCT:

- id int
- name varchar(40)
- productionCost decimal(10,2)
- idCategory int
- PK (id)
- FK (idCategory) ref. Category(id)

#### • SALES:

- salesNumber int
- orderDate date
- idClient int
- address varchar(100)
- idCity int
- PK(salesNumber)
- FK (idClient) ref Client(id)
- FK (idCity) ref City(id)
- UQ(idClient,orderDate)

#### • SALESDetails:

- salesOrderNumber int
- salesNumber int
- orderQuantity int
- idProduct int
- salesAmount (campo calcolato, si userà una view)
- PK (salesNumber,salesOrderNumber)
- FK (salesNumber) ref Sales(salesNumber)
- FK (idProduct) ref. Product(idProduct)

## DESCRIZIONE DATABASE GENERATO

La creazione delle tabelle è stato eseguito seguendo lo schema logico illustrato precedentemente.

L'inserimento dei dati è stato agevolato grazie all'aiuto di ChatGPT al quale è stato richiesto esclusivamente di generare i dati da utilizzare per le query richieste. L'uso di ChatGPT è stato consigliato direttamente dal docente durante la spiegazione della consegna dell'esame per agevolare e ridurre i tempi di popolazione del database.

Le categorie nel DB sono 5.

Le regioni nel DB sono 7.

Le città nel DB sono 70, 10 per ogni regione.

I clienti nel DB sono 31, 10 dei quali non hanno fatto ordini.

I prodotti sono 50, 5 dei quali sono invenduti.

Gli ordini (sales) sono 156 generando 631 record in salesDetails.

## POSSIBILE ERRORE

```
91  /*2. Esporre l'elenco dei soli prodotti venduti e per ognuno di questi il fatturato totale per anno.*/
92
93  SELECT p.`productName`
94         , fatturatoPerAnno.annoDiRiferimento
95         , fatturatoPerAnno.totaleFatturato
96  FROM(
97      SELECT sd.`idProduct` as idProdotto
98             , YEAR(s.`orderDate`) as annoDiRiferimento
99             , SUM (vw.`salesAmount`) as totaleFatturato
100  FROM sales s
101      JOIN salesdetails sd ON s.`salesNumber` = sd.`salesNumber`
102      JOIN vw_salesamount vw ON vw.`salesNumber` = sd.`salesNumber` AND vw.`salesLineNumber` = sd.`salesLineNum
103  GROUP BY sd.`idProduct`,YEAR(s.`orderDate`)
104  ORDER BY sd.`idProduct`,YEAR(s.`orderDate`)) as fatturatoPerAnno
105  JOIN product p ON fatturatoPerAnno.idProdotto = p.`idProduct`;
106
```

00% 67:105

Action Response

1 SELECT p.`productName` , fatturatoPerAnno.annoDiRiferime... Error Code: 1630. FUNCTION toysgroup\_napakristian.SUM does not exist. Check the 'Function Name Parsing

Eseguendo la query su Workbench mi restituisce errore nel SUM, dicendo che nel database creato non esiste la funzione SUM.  
Di seguito allego l'output della query eseguita con VSCode nel quale l'errore non si presenta

The screenshot shows the VS Code interface with a SQL query editor on the right and a results table on the left. The query is the same as the one in the first image, but it runs successfully in VS Code. The results table shows the following data:

productName	annoDiRiferimento	totaleFatturato
Laptop	2019	6589.00
Smartwatch	2019	2864.00
Smartwatch	2020	895.00
Televisore	2019	4500.00
Televisore	2020	2250.00
Cuffie Bluetooth	2019	3293.00
Cuffie Bluetooth	2020	1691.00
Powerbank	2019	374.00
Powerbank	2020	1836.00
Speaker Wireless	2019	5282.00
Sneaker Wireless	2020	9313.00