

Le **Viste** logiche, oltre ad essere utilizzate per fornire ad ogni utente le sole informazioni di suo interesse o i dati a cui è abilitato, sono utili per scrivere interrogazioni complesse che richiederebbero subQuery poco leggibili oppure per evitare di scrivere più volte parti di una query già utilizzata.

Ad esempio, per le ultime 3 interrogazioni esposte nel documento sulle subQuery:

- 1) esporre per ogni cliente gli importi degli ordini evasi (status = 'shipped')
- 2) determinare per ogni cliente gli ordini evasi di massimo importo
- 3) determinare per l'anno 2004 e per ogni paese (country) l'importo totale degli ordini evasi

dove per le ultime 2 si è utilizzata come query interna la prima, si può **definire ed utilizzare una Vista**.

Definizione della **vista** logica:

CREATE VIEW V_cust_details AS

```
SELECT country, city, customerNumber, customerName, orderNumber, orderDate,
       SUM(quantityOrdered * priceEach) AS importo
FROM customers INNER JOIN orders USING (customerNumber)
      INNER JOIN orderdetails USING (orderNumber)
WHERE status = 'shipped'
GROUP BY country, city, customerNumber, orderNumber;
```

Utilizzo della vista (come fosse una tabella) per ottenere i 3 risultati richiesti:

Query 1) - 303 righe

```
SELECT *
FROM V_cust_details
ORDER BY city, importo
```

country	city	customerNumber	customerName	orderNumber	orderDate	importo
USA	Allentown	157	Diecast Classics Inc.	10272	2004-07-20	23715.70
USA	Allentown	157	Diecast Classics Inc.	10318	2004-11-02	35152.12
USA	Allentown	157	Diecast Classics Inc.	10281	2004-08-19	39641.43
New Zealand	Auckland	323	Down Under Souveniers, Inc	10132	2003-06-25	2880.00
New Zealand	Auckland	357	GiftsForHim.com	10202	2003-12-02	20220.04
New Zealand	Auckland	496	Kelly's Gift Shop	10399	2005-04-01	30253.75
...

Query 2) - 98 righe

```
SELECT country, city, customerNumber, customerName, MAX(importo) AS impMAX
FROM V_cust_details
GROUP BY country, city, customerNumber
ORDER BY city, impMAX
```

country	city	customerNumber	customerName	impMAX
USA	Allentown	157	Diecast Classics Inc.	39641.43
New Zealand	Auckland	357	GiftsForHim.com	36442.34
New Zealand	Auckland	323	Down Under Souveniers, Inc	41426.81
...

Query 3) - 20 righe

```
SELECT country, YEAR(orderDate) AS anno, COUNT(customerNumber) AS numCli,
       COUNT(orderNumber) AS ordini, SUM(importo) AS impTOT
FROM V_cust_details
WHERE YEAR(orderDate) = 2004
GROUP BY country, anno
ORDER BY impTOT DESC
```

country	anno	numCli	ordini	impTOT
USA	2004	52	52	1485054.44
France	2004	19	19	506660.01
Spain	2004	13	13	392816.48
Australia	2004	6	6	204213.18
New Zealand	2004	5	5	195592.89
...

Altri esempi:

Determinare i valori minimi, massimi e medi del prezzo di vendita e delle quantità ordinate.

Definizione vista logica:

```
CREATE VIEW V_stats_details AS
SELECT MIN(priceEach) AS przMIN,
       MAX(priceEach) AS przMAX,
       ROUND(AVG(priceEach), 2) AS przMED,
       MIN(quantityOrdered) AS qtaMIN,
       MAX(quantityOrdered) AS qtaMAX,
       ROUND(AVG(quantityOrdered), 2) AS qtaMED
FROM orderdetails
```

Utilizzo della vista

```
SELECT *
FROM V_stats_details
```

przMIN	przMAX	przMED	qtaMIN	qtaMAX	qtaMED
26.55	214.30	90.77	6	97	35.22

Individuare i prodotti venduti al maggior e al minor prezzo e il prodotto venduto al prezzo più vicino a quello medio, esponendo anche il rispettivo guadagno unitario (per singolo pezzo)

```
SELECT DISTINCT productName, productLine, buyPrice, priceEach, priceEach - buyPrice AS guadagno
FROM orderdetails INNER JOIN products USING (productCode)
```

```
WHERE priceEach = (SELECT przMAX FROM V_stats_details)
OR priceEach = (SELECT przMIN FROM V_stats_details)
OR priceEach >= (SELECT przMED FROM V_stats_details) - 0.02
AND priceEach <= (SELECT przMED FROM V_stats_details) + 0.02
-- invece di (SELECT MAX(priceEach) FROM orderdetails) e simili come query interne
```

productName	productLine	buyPrice	priceEach	guadagno
1952 Alpine Renault 1300	Classic Cars	98.58	214.30	115.72
America West Airlines B757-200	Planes	68.80	90.75	21.95
1939 Chevrolet Deluxe Coupe	Vintage Cars	22.57	26.55	3.98

Individuare le linee di prodotto acquistate con un ordine di quantità pari alla massima, minima e media (determinando anche il numero di ordini con tali valori)

```
SELECT productLine, quantityOrdered AS qtaOrd,
       COUNT(quantityOrdered) AS num -- oppure COUNT(orderNumber)
```

```
FROM orderdetails INNER JOIN products USING (productCode)
WHERE quantityOrdered = (SELECT qtaMAX FROM V_stats_details)
OR quantityOrdered = (SELECT qtaMIN FROM V_stats_details)
OR quantityOrdered >= (SELECT qtaMED FROM V_stats_details) - 0.6
AND quantityOrdered <= (SELECT qtaMED FROM V_stats_details) + 0.6
```

```
GROUP BY productLine, quantityOrdered
ORDER BY quantityOrdered
```

productLine	qtaOrd	num
Vintage Cars	6	2
Trucks and Buses	35	9
Vintage Cars	35	17
Classic Cars	35	20
Motorcycles	35	11
Planes	35	10
Trains	35	3
Ships	35	7
Classic Cars	97	1

Individuare per ogni prodotto il numero degli ordini inseriti, la quantità totale ordinata e tutti i valori economici (valore minimo, massimo e medio del prezzo di vendita, la spesa, i ricavi e i guadagni).

Definizione **vista logica**:

```

CREATE VIEW V_stats_details_Xprod AS
SELECT    productCode , productName , productLine , buyPrice , quantityInStock ,
          COUNT(orderNumber)                AS numOrd,
          MIN(priceEach)                     AS przMIN,
          MAX(priceEach)                     AS przMAX,
          ROUND(AVG(priceEach), 2)           AS przMED,
          SUM(quantityOrdered)               AS qtaTOT,
          MIN(quantityOrdered)               AS qtaMIN,
          MAX(quantityOrdered)               AS qtaMAX,
          ROUND(AVG(quantityOrdered), 2)     AS qtaMED,
          SUM(quantityOrdered * priceEach)   AS ricavoTOT,
          SUM(quantityOrdered) * buyPrice    AS spesaTOT ,
          SUM(quantityOrdered * (priceEach-buyPrice)) AS guadagno,
          buyPrice * quantityInStock         AS valoreStock
FROM      orderdetails RIGHT JOIN products USING (productCode)
GROUP BY productCode

```

Utilizzo della **vista**

```

SELECT * FROM V_stats_details_Xprod

```

Individuare il prodotto più venduto e quello meno richiesto, quelli che hanno determinato il massimo e il minimo guadagno, quelli di min. e massimo prezzo unitario e quello per il quale si è speso di meno.

```

SELECT productCode , productName , productLine , qtaTOT , guadagno , spesaTOT , buyPrice
FROM V_stats_details_Xprod
WHERE qtaTOT    = (SELECT MAX(qtaTOT) FROM V_stats_details_Xprod)  -- 1
OR qtaTOT      = (SELECT MIN(qtaTOT) FROM V_stats_details_Xprod)  -- 2
OR guadagno    = (SELECT MAX(guadagno) FROM V_stats_details_Xprod) -- 3
OR guadagno    = (SELECT MIN(guadagno) FROM V_stats_details_Xprod) -- 4
OR spesaTOT    = (SELECT MIN(spesaTOT) FROM V_stats_details_Xprod) -- 5
OR buyPrice    = (SELECT MIN(buyPrice) FROM V_stats_details_Xprod) -- 6
OR buyPrice    = (SELECT MAX(buyPrice) FROM V_stats_details_Xprod) -- 7
ORDER BY guadagno

```

productCode	productName	productLine	qtaTOT	guadagno	spesaTOT	buyPrice	condizione
S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	937	6904.85	21148.09	22.57	4
S24_2840	1958 Chevy Corvette Limited Edition	Classic Cars	983	15988.43	15639.53	15.91	6
S24_2972	1982 Lamborghini Diablo	Classic Cars	912	16161.99	14810.88	16.24	5
S18_4933	1957 Ford Thunderbird	Classic Cars	767	23862.50	26239.07	34.21	2
S10_4962	1962 LanciaA Delta 16V	Classic Cars	932	26735.57	96387.44	103.42	7
S18_3232	1992 Ferrari 360 Spider red	Classic Cars	1808	135996.78	140843.20	77.90	1 e 3

Anche le **Viste logiche** possono essere usate nelle JOIN, così come le **tabelle fisiche**.

Esempi:

Individuare i prodotti venduti al minimo prezzo in assoluto (**przMIN** in **V_stats_details**) ed esporre nome, linea di prodotto, **prezzo di acquisto**, prezzo minimo, massimo e medio di vendita del prodotto (su **V_stats_details_Xprod**), prezzo massimo e medio in assoluto (su **V_stats_details**).

```
SELECT productCode , productName , productLine , buyPrice , przMIN,
       VXP.przMAX,                VXP.przMED,
       VA.przMAX AS prMAXas,      VA.przMED AS prMEDas
FROM   V_stats_details_Xprod VXP
INNER JOIN V_stats_details VA   USING (przMIN)
```

productCode	productName	productLine	buyPrice	przMIN	przMAX	przMED	prMAXas	prMEDas
S24_1937	1939 Chevrolet Deluxe Coupe	Vintage Cars	22.57	26.55	33.19	29.93	214.30	90.77

Individuare i prodotti venduti con dettagli di quantità pari al minimo assoluto (**qtaMIN** in **V_stats_details**) ed esporre nome, linea di prodotto, **quantità totale**, q. massima, minima e media di vendita del prod. (su **V_stats_details_Xprod**), quantità massima e media assolute (su **V_stats_details**).

```
SELECT productCode , productName , productLine , qtaTOT, qtaMIN,
       VXP.qtaMAX,                VXP.qtaMED,
       VA.qtaMAX AS qtaMAXas,     VA.qtaMED AS qtaMEDas
FROM   V_stats_details_Xprod VXP
INNER JOIN V_stats_details VA   USING (qtaMIN)
```

productCode	productName	productLine	qtaTOT	qtaMIN	qtaMAX	qtaMED	qtaMAXas	qtaMEDas
S18_2325	1932 Model A Ford J-Coupe	Vintage Cars	957	6	50	34.18	97	35.22
S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars	866	6	66	34.64	97	35.22

Individuare i clienti che ha effettuato ordini con dettagli di quantità pari al max assoluto (**qtaMAX** in **V_stats_details**), esporre nome e nazionalità e numeri d'ordine, prodotto, linea di prodotto e quantità.

```
SELECT customerName, country, orderNumber, orderLineNumber AS lineN,
       VXP.productCode, productName, productLine, qtaMAX
FROM   V_stats_details_Xprod VXP
INNER JOIN V_stats_details VA USING (qtaMax)
INNER JOIN orderdetails O ON VXP.productCode = O.productCode
           AND quantityOrdered = qtaMAX
INNER JOIN orders USING (orderNumber)
INNER JOIN customers USING (customerNumber)
```

customerName	country	orderNumber	lineN	productCode	productName	productLine	qtaMAX
Mini Caravy	France	10405	5	S12_4675	1969 Dodge Charger	Classic Cars	97