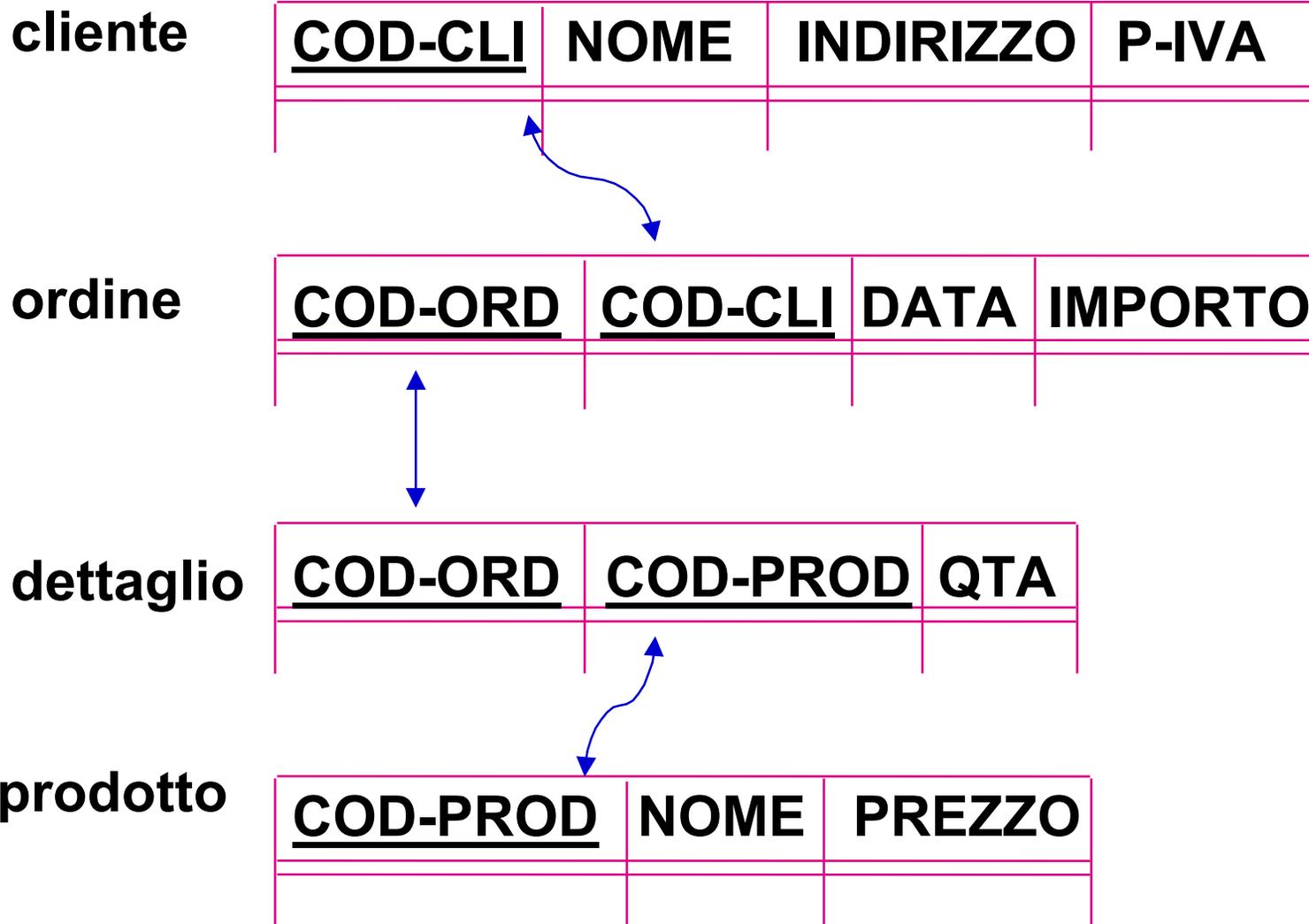


# Interrogazioni complesse

# Classificazione delle interrogazioni complesse

- **Query con ordinamento**
- **Query con aggregazione**
- **Query con raggruppamento**
- **Query binarie**
- **Query annidate**

# Esempio : gestione ordini



# Istanza di ordine

ordine

<b>COD-ORD</b>	<b>COD-CLI</b>	<b>DATA</b>	<b>IMPORTO</b>
<b>1</b>	<b>3</b>	<b>1997-06-01</b>	<b>50.000.000</b>
<b>2</b>	<b>4</b>	<b>1997-08-03</b>	<b>8.000.000</b>
<b>3</b>	<b>3</b>	<b>1997-09-01</b>	<b>5.500.000</b>
<b>4</b>	<b>1</b>	<b>1997-07-01</b>	<b>12.000.000</b>
<b>5</b>	<b>1</b>	<b>1997-08-01</b>	<b>1.500.000</b>
<b>6</b>	<b>3</b>	<b>1997-09-03</b>	<b>27.000.000</b>

# Query con ordinamento

```
SELECT * FROM ORDINE  
WHERE IMPORTO > 100.000 ORDER BY DATA
```

<b>COD-ORD</b>	<b>COD-CLI</b>	<b>DATA</b>	<b>IMPORTO</b>
<b>1</b>	<b>3</b>	<b>1997-06-01</b>	<b>50.000.000</b>
<b>4</b>	<b>1</b>	<b>1997-07-01</b>	<b>12.000.000</b>
<b>5</b>	<b>1</b>	<b>1997-08-01</b>	<b>1.500.000</b>
<b>2</b>	<b>4</b>	<b>1997-08-03</b>	<b>8.000.000</b>
<b>3</b>	<b>3</b>	<b>1997-09-01</b>	<b>5.500.000</b>
<b>6</b>	<b>3</b>	<b>1997-09-03</b>	<b>27.000.000</b>

# Order by

```
SELECT * FROM ORDINE  
WHERE IMPORTO > 100.000  
ORDER BY COD-CLI
```

COD-ORD	COD-CLI	DATA	IMPORTO
4	1	1997-07-01	12.000.000
5	1	1997-08-01	1.500.000
1	3	1997-06-01	50.000.000
6	3	1997-09-03	5.500.000
3	3	1997-09-01	27.000.000
2	4	1997-08-03	8.000.000

# Order by

**Nel caso si voglia ordinare sui valori di una colonna risultato di una espressione si usa una notazione “posizionale”**

```
SELECT COD-CLI, IMPORTO/1.000.000  
FROM ORDINE  
WHERE IMPORTO > 100.000  
ORDER BY 2
```

**(ordina sulla SECONDA colonna)**

# Order by

.....

**ORDER BY COD-CLI ASC, DATA DESC**

<b>COD-ORD</b>	<b>COD-CLI</b>	<b>DATA</b>	<b>IMPORTO</b>
<b>5</b>	<b>1</b>	<b>1997-08-01</b>	<b>1.500.000</b>
<b>4</b>	<b>1</b>	<b>1997-07-01</b>	<b>12.000.000</b>
<b>6</b>	<b>3</b>	<b>1997-09-03</b>	<b>27.000.000</b>
<b>3</b>	<b>3</b>	<b>1997-09-01</b>	<b>5.500.000</b>
<b>1</b>	<b>3</b>	<b>1997-06-01</b>	<b>50.000.000</b>
<b>2</b>	<b>4</b>	<b>1997-08-03</b>	<b>8.000.000</b>

# Query aggregate

Utilizzano le funzioni aggregate:

**SUM**      sommatoria

**AVG**      media

**MIN**      minimo

**MAX**      massimo

**COUNT**    cardinalità

**(I NULL sono esclusi)**

# Query con massimo

- Selezionare l'importo massimo degli ordini.

```
SELECT MAX(IMPORTO) AS MAX-IMP  
FROM ORDINE
```

<b>MAX-IMP</b>
<b>50.000.000</b>

# Query con sommatoria

- Selezionare la somma degli importi degli ordini relativi al cliente numero 1.

```
SELECT SUM(IMPORTO) AS SOMMA-IMP  
FROM ORDINE  
WHERE COD-CLIENTE = 1
```

<b>SOMMA-IMP</b>
<b>13.500.000</b>

# Query con raggruppamento

si aggiungono le clausole

**GROUP-BY** (raggruppamento)

**HAVING** (selezione dei gruppi)

**SELECT** .....

**FROM** .....

**WHERE** .....

**GROUP BY** .....

**HAVING** .....

# Query con raggruppamento

- **Selezionare la somma degli importi degli ordini successivi al 10-6-97 per quei clienti che hanno emesso almeno 2 ordini.**

```
SELECT COD-CLI, SUM (IMPORTO)  
FROM ORDINE  
WHERE DATA > '1997-06-10'  
GROUP BY COD-CLI  
HAVING COUNT (IMPORTO) >= 2
```

# Passo 1: Valutazione WHERE

<b>COD-ORD</b>	<b>COD-CLI</b>	<b>DATA</b>	<b>IMPORTO</b>
<b>2</b>	<b>4</b>	<b>1997-08-03</b>	<b>8.000.000</b>
<b>3</b>	<b>3</b>	<b>1997-09-01</b>	<b>5.500.000</b>
<b>4</b>	<b>1</b>	<b>1997-07-01</b>	<b>12.000.000</b>
<b>5</b>	<b>1</b>	<b>1997-08-01</b>	<b>1.500.000</b>
<b>6</b>	<b>3</b>	<b>1997-09-03</b>	<b>27.000.000</b>

**( WHERE DATA > '1997-06-10' )**

# Passo 2 : Raggruppamento

si valuta la clausola **GROUP-BY**

<b>COD-ORD</b>	<b>COD-CLI</b>	<b>DATA</b>	<b>IMPORTO</b>
4	1	1997-07-01	12.000.000
5	1	1997-08-01	1.500.000
3	3	1997-09-01	5.500.000
6	3	1997-09-03	27.000.000
2	4	1997-08-03	8.000.000

**(GROUP BY COD-CLI)**

# Passo 3 : Calcolo degli aggregati

per ciascun gruppo si calcolano  
**SUM (IMPORTO)** e **COUNT (IMPORTO)**

<b>COD-CLI</b>	<b>SUM (IMPORTO)</b>	<b>COUNT (IMPORTO)</b>
<b>1</b>	<b>13.500.000</b>	<b>2</b>
<b>3</b>	<b>32.500.000</b>	<b>2</b>
<b>4</b>	<b>8.000.000</b>	<b>1</b>

# Passo 4 : Selezione dei gruppi

si valuta il predicato

**COUNT (IMPORTO) >=2**

<b>COD-CLI</b>	<b>SUM (IMPORTO)</b>	<b>COUNT (IMPORTO)</b>
<b>1</b>	<b>13.500.000</b>	<b>2</b>
<b>3</b>	<b>32.500.000</b>	<b>2</b>
<b>4</b>	<b>8.000.000</b>	<b>1</b>

# Passo 5 : Produzione del risultato

<b>COD-CLI</b>	<b>SUM (IMPORTO)</b>
<b>1</b>	<b>13.500.000</b>
<b>3</b>	<b>32.500.000</b>

# Doppio raggruppamento

- **Selezionare la somma delle quantità dei dettagli degli ordini emessi da ciascun cliente per ciascun prodotto, purché la somma superi 50.**

```
SELECT COD-CLI, COD-PROD, SUM(QTA)  
FROM ORDINE AS O, DETTAGLIO AS D  
WHERE O.COD-ORD = D.COD-ORD  
GROUP BY COD-CLI, COD-PROD  
HAVING SUM(QTA) > 50
```

# Situazione dopo il join e il raggruppamento

ordine

dettaglio

COD-CLI	ORDINE. COD-ORD	DETTAGLIO. COD-ORD	COD-PROD	QTA
1	3	3	1	30
1	4	4	1	20
1	3	3	2	30
1	5	5	2	10
2	3	3	1	60
3	1	1	1	40
3	2	2	1	30
3	6	6	1	25

gruppo 1,1

gruppo 1,2

gruppo 2,1

gruppo 3,1

# Estrazione del risultato

si valuta la funzione aggregata **SUM(QTA)** e il predicato **HAVING**

COD-CLI	COD-PROD	SUM(QTA)
1	1	50
1	2	40
2	1	60
3	1	95

# Query con raggruppamento e ordinamento

- E' possibile ordinare il risultato anche delle query con raggruppamento

```
SELECT .....  
FROM .....  
[ WHERE ..... ]  
GROUP BY .....  
[ HAVING ..... ]  
ORDER BY ....
```

# Raggruppamento e ordinamento

- Selezionare la somma degli importi degli ordini successivi al 10/6/97 per quei clienti che hanno emesso almeno 2 ordini dopo quella data, in ordine di cliente.

```
SELECT COD-CLI, SUM (IMPORTO)  
FROM ORDINE  
WHERE DATA > 1997-06-10  
GROUP BY COD-CLI  
HAVING COUNT (IMPORTO) >= 2  
ORDER BY COD-CLI
```

# ESERCIZI SUL GROUP BY

1     **SELECT ART-NO, SPED-NO, NUM**  
      **FROM SPEDIZIONI**  
      **GROUP BY ART-NO**

**OPPURE**

2     **SELECT ART-NO, SPED-NO, NUM**  
      **FROM SPEDIZIONI**  
      **GROUP BY ART-NO,SPED-NO**

# ESERCIZI SUL GROUP BY

**SU:**

<b>ART-NO</b>	<b>SPED-NO</b>	<b>NUM</b>	<b>(SPEDIZIONI)</b>
1	2	100	
1	2	25	
1	3	75	
2	2	100	
2	4	100	
2	4	75	
1	4	25	

# ESERCIZI SUL GROUP BY

## 1 ....GROUP BY ART-NO

ART-NO	SPED-NO	NUM
1	2,2,3,4	100,25,75,25
2	2,4,4	100,100,75

## 1 ....GROUP BY ART-NO, SPED-NO

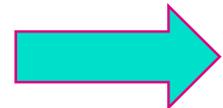
ART-NO	SPED-NO	NUM
1	2	100,25
1	3	75
1	4	25
2	2	100
2	4	100,75

# ESERCIZI SUL GROUP BY

```
SELECT ARTICOLO  
FROM VENDITE GROUP BY ARTICOLO  
HAVING COUNT(DISTINCT DATA)
```

SU

ARTICOLO	DATA	NUM
PENNA	1997-12-13	2
MATITA	1997-12-08	1
GOMMA	1997-12-10	1
MATITA	1997-12-08	2
PENNA	1997-12-08	1



# ESERCIZI SUL GROUP BY

## GROUP BY ARTICOLO

SU

**ARTICOLO**

**DATA**

**PENNA**

**1997-12-13, 1997-12-08**

**MATITA**

**1997-12-08, 1997-12-08**

**GOMMA**

**1997-12-10**

## HAVING COUNT(DISTINCT DATA)

**ARTICOLO**

**PENNA**

# Query binarie

**Costruite concatenando due query SQL tramite operatori insiemistici:**

<b>UNION</b>	<b>unione</b>
<b>INTERSECT</b>	<b>intersezione</b>
<b>EXCEPT (MINUS)</b>	<b>differenza</b>

**(si eliminano i duplicati)**

# Unione

- **Selezionare i codici degli ordini i cui importi superano 500.000 lire oppure presenti in qualche dettaglio con quantità superiore a 1000.**

```
SELECT COD-ORD  
FROM ORDINE  
WHERE IMPORTO > 500.000  
UNION  
SELECT COD-ORD  
FROM DETTAGLIO  
WHERE QTA > 1000
```

# Differenza

- **Selezionare i codici degli ordini i cui importi superano 500.000 lire ma non presenti in nessun dettaglio con quantità superiore a 1000.**

```
SELECT COD-ORD  
FROM ORDINE  
WHERE IMPORTO > 500.000  
EXCEPT  
SELECT COD-ORD  
FROM DETTAGLIO  
WHERE QTA > 1000
```

# Intersezione

- **Selezionare i codici degli ordini i cui importi superano 500.000 lire e che sono presenti in qualche dettaglio con quantità superiore a 1000.**

```
SELECT COD-ORD  
FROM ORDINE  
WHERE IMPORTO > 500.000  
INTERSECT  
SELECT COD-ORD  
FROM DETTAGLIO  
WHERE QTA > 1000
```

# Query annidate

**Costruite concatenando due query SQL nel predicato where:**

<b>[NOT] IN</b>	<b>appartenenza</b>
<b>[DOES NOT] EXISTS</b>	<b>esistenza</b>
<b>ANY, ALL</b>	<b>quantificatori</b>

**comparatore: =, !=, <, <=, >, >=**

# Query nidificate (nested)

- Selezionare nome e indirizzo dei clienti che hanno emesso qualche ordine di importo superiore a 10.000.000.

```
SELECT NOME, INDIRIZZO
FROM CLIENTE
WHERE COD-CLI IN
    ( SELECT COD-CLI
      FROM ORDINE
      WHERE IMPORTO > 10.000.000 )
```

# Equivalenza fra IN e query semplici

```
SELECT NOME, INDIRIZZO
FROM CLIENTE
WHERE COD-CLI IN
  ( SELECT COD-CLI
    FROM ORDINE
    WHERE IMPORTO > 10.000.000 )
```

**equivale (a meno di duplicati) a:**

```
SELECT NOME, INDIRIZZO
FROM CLIENTE JOIN ORDINE
  ON CLIENTE.COD-CLI = ORDINE.COD-CLI
WHERE IMPORTO > 10.000.000
```

# Nested Query complesse (fino a 8 livelli)

- Selezionare nome e indirizzo dei clienti che hanno emesso qualche ordine i cui dettagli comprendono il prodotto “Pneumatico”.

```
SELECT NOME, INDIRIZZO FROM CLIENTE
WHERE COD-CLI IN
  ( SELECT COD-CLI FROM ORDINE
    WHERE COD-ORD IN
      ( SELECT COD-ORD FROM DETTAGLIO
        WHERE COD-PROD IN
          ( SELECT COD-PROD FROM PRODOTTO
            WHERE NOME = 'Pneumatico' ) ) ) )
```

# La query equivalente

equivale (a meno di duplicati) a:

```
SELECT NOME, INDIRIZZO  
FROM CLIENTE AS C, ORDINE AS O,  
    DETTAGLIO AS D, PRODOTTO AS P  
WHERE C.COD-CLI = O.COD-CLI  
    AND O.COD-ORD = D.COD-ORD  
    AND D.COD-PROD = P.COD-PROD  
    AND NOME= 'Pneumatico'
```

# Nested query correlate (con variabile)

```
SELECT NOME
FROM IMPIEGATI AS X,
WHERE SALARIO >
  ( SELECT SALARIO
    FROM IMPIEGATI
    WHERE MATRICOLA = X.MANAGER )
```

sulla relazione IMPIEGATI (matricola, nome, manager...)  
equivale al JOIN

```
SELECT X.NOME
FROM IMPIEGATI AS X, IMPIEGATI AS Y
WHERE X.SALARIO > Y.SALARIO
AND Y.MATRICOLA = X.MANAGER
```

# Uso di IN nelle modifiche

- aumentare di L. 5000 l'importo di tutti gli ordini che comprendono il prodotto 456

```
UPDATE ORDINE
SET IMPORTO = IMPORTO + 5000
WHERE COD-ORD IN
  ( SELECT COD-ORD
    FROM DETTAGLIO
    WHERE COD-PROD = '456' )
```

# Query con NOT IN

- **Selezionare nome e indirizzo dei clienti che non hanno emesso nessun ordine di importo superiore a 10.000.000.**

```
SELECT NOME, INDIRIZZO  
FROM CLIENTE  
WHERE COD-CLI NOT IN  
    ( SELECT COD-CLI  
      FROM ORDINE  
      WHERE IMPORTO > 10.000.000 )
```

# La query con ANY e ALL

**SELECT COD-ORD  
FROM ORDINE  
WHERE IMPORTO > ANY  
( SELECT IMPORTO  
FROM ORDINE )**

<b>COD-ORD</b>	<b>IMPORTO</b>
<b>1</b>	<b>50.000</b>
<b>2</b>	<b>300.000</b>
<b>3</b>	<b>90.000</b>

**SELECT COD-ORD  
FROM ORDINE  
WHERE IMPORTO >= ALL  
( SELECT IMPORTO  
FROM ORDINE )**

<b>ANY</b>	<b>ALL</b>
<b>F</b>	<b>F</b>
<b>V</b>	<b>V</b>
<b>V</b>	<b>F</b>

# Esercizi

- **Riprendere le basi di dati per la gestione del personale e degli ordini ed esprimere in SQL le interrogazioni :**
  - **quale impiegati lavorano in un progetto in cui non lavora il loro manager?**
  - **quanti ordini ha emesso Paolo?**
  - **quante candele sono state ordinate il 5/7/97?**
  - **calcolare per ciascun cliente la somma degli importi di tutti gli ordini**
  - **estrarre l'ordine di importo più alto**

# Esercizi (lezione precedente)

- in quali tipi di progetti lavora Giovanni?

```
SELECT TIPO FROM PROGETTO
WHERE NUM-PROG IN
  ( SELECT NUM-PROG FROM ASSEGNAAMENTO
    WHERE MATR IN
      ( SELECT MATR FROM IMPIEGATO
        WHERE NOME='Giovanni' ) )
```

- chi è il manager di Piero?

```
SELECT NOME FROM IMPIEGATO
WHERE MATR IN
  ( SELECT MATR -MGR FROM IMPIEGATO
    WHERE NOME='Piero' )
```

# Query annidate (nested)

- **Selezionare nome e indirizzo dei clienti che hanno emesso qualche [o non hanno emesso nessun] ordine di importo superiore a 10.000.000.**

```
SELECT NOME, INDIRIZZO  
FROM CLIENTE C  
WHERE [NOT] EXISTS  
( SELECT *  
  FROM ORDINE  
  WHERE IMPORTO > 10.000.000  
    AND COD_CLI=C.COD_CLI )
```