

# Le cose che **non** vogliamo vedere

- Clausole where auto-contraddittorie
- IN / NOT IN con
  - Niente a sinistra `where [not] in ( select ... )`
  - Schemi che non si corrispondono  
`where DataInizio [not] in ( select Nome, Cognome ... )`
- Predicati con query a destra e a sinistra. Es:  
`Where (Select ... FROM ...) > (Select ... From ...)`
- Da usare con consapevole parsimonia (*non sono errori*):
  - Predicati con query nidificate a destra senza ANY/ALL
  - EXISTS (SELECT X...) con X != \*

# Interpretazione

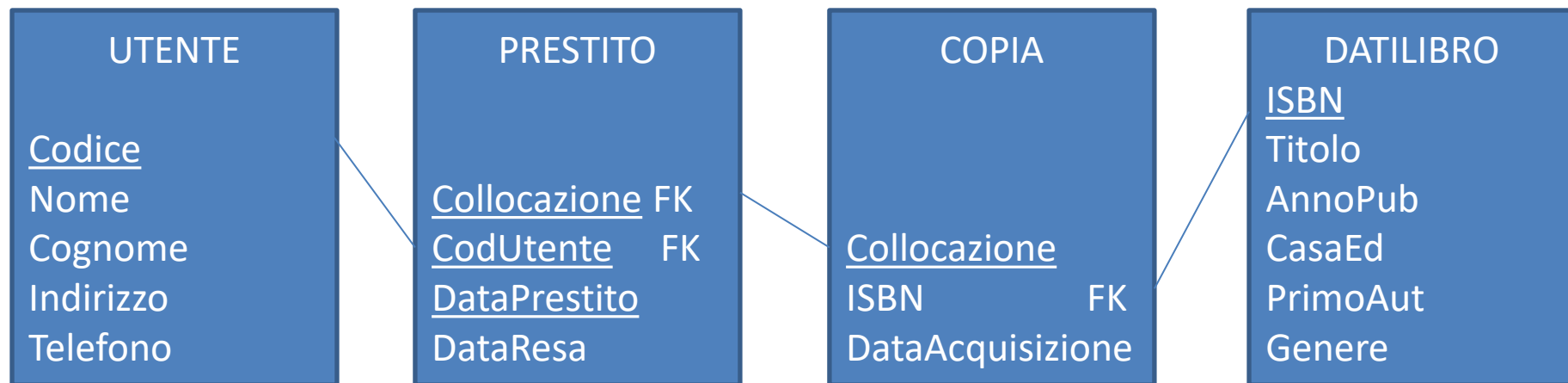
- ```
select CodCli
from Ordine O
where exists (select *
              from Ordine O1
              where O1.CodCli = O.CodCli
                 and O1.Data = O.Data
                 and O1.CodOrd <> O.CodOrd)
```
- Per ogni tupla di Ordine:
  - Si passano alla query interna CodCli, Data, CodOrd
  - Si valuta la query interna
  - Se la query non è vuota il CodCli di quella tupla è proiettato nel risultato

# Equivalenza del potere espressivo

- IN, =ANY, EXISTS hanno lo stesso potere espressivo e possono essere espresse tramite join (a meno di duplicati)
- NOT IN, <>ALL, NOT EXISTS hanno lo stesso potere espressivo e possono essere rese tramite la differenza
- <comp> ANY, a meno di duplicati, può essere reso con un theta-join (non equi-join)
- <comp> ALL può essere reso con query che abbiano raggruppamento e estrazione di minimo o massimo

UTENTE(Codice, Nome, Cognome, Indirizzo, Telefono)  
PRESTITO(Collocazione, CodUtente, DataPrestito, DataResa)  
COPIA(Collocazione, ISBN, DataAcquisizione)  
DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

**N.B. :** Nel seguito, per «motivi di impaginazione», alcuni attributi possono essere abbreviati (ad esempio: Collocazione → Colloc → Coll), senza mai tuttavia diventare ambigui.



UTENTE(Codice, Nome, Cognome, Indirizzo, Telefono)

PRESTITO(Collocazione, CodUtente, DataPrestito, DataResa)

COPIA(Collocazione, ISBN, DataAcquisizione)

DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

5) *Individuare, per ogni genere, il libro acquisito per primo dalla biblioteca*

*Si può esprimere «direttamente» usando il raggruppamento ? :*

```
select Genere, min(DataAcquisizione), Titolo, Coll  
from Copia C join DatiLibro D on C.Isbn = D.Isbn  
group by Genere
```

**NO !!!**

UTENTE(Codice, Nome, Cognome, Indirizzo, Telefono)

PRESTITO(Collocazione, CodUtente, DataPrestito, DataResa)

COPIA(Collocazione, ISBN, DataAcquisizione)

DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

5) *Individuare, per ogni genere, il libro acquisito per primo dalla biblioteca*

```
select Genere, DataAcquisizione, Titolo, Collocazione
from Copia C join DatiLibro D on C.Isbn = D.Isbn
where DataAcquisizione =
    ( select min(DataAcquisizione)
      from Copia C2 join DatiLibro D2 on C2.Isbn=D2.Isbn
      where D2.Genere = D.Genere )
```

5) *Individuare, per ogni genere, il libro acquisito per primo dalla biblioteca*

*Usando il raggruppamento e il costruttore di tupla :*

```
select Genere, DataAcquisizione, Titolo, Collocazione
from Copia C join DatiLibro D on C.Isbn = D.Isbn
where (DataAcquisizione, Genere) in
    ( select min(DataAcquisizione), Genere
      from Copia C join DatiLibro D on C.Isbn = D.Isbn
      group by Genere )
```

5) *Individuare, per ogni genere, il libro acquisito per primo dalla biblioteca*

**CON UNA VISTA :**

```
create view PrimoDiOgniGen( Date, Genre ) as
( select min(DataAcquisizione), Genre
  from Copia C join DatiLibro D on C.Isbn = D.Isbn
  group by Genre )

select Genre, DataAcquisizione, Titolo, Collocazione
from Copia C join DatiLibro D
  on C.Isbn = D.Isbn
  join PrimoDiOgniGen
    on (Genre,DataAcquisizione)=(Genre,Date)
```



5) Individuare, *per ogni genere*, il libro acquisito per primo dalla biblioteca

Usando il costruttore di tupla e la vista definita prima :

```
create view PrimoDiOgniGen( Date, Genre ) as
( select min(DataAcquisizione), Genre
  from Copia C join DatiLibro D on C.Isbn = D.Isbn
  group by Genre )

select Genre, DataAcquisizione, Titolo, Collocazione
from Copia C join DatiLibro D on C.Isbn = D.Isbn
where (DataAcquisizione,Genre) in ( select *
                                   from PrimoDiOgniGen )
```

6) Quando è stato pubblicato il libro **giallo** più vecchio (isbn) che sia **di una** casa editrice di cui la biblioteca **ha** almeno 100 diversi libri (isbn) ?

```
select min( AnnoPub )  
from DatiLibro  
where Genere = "Giallo"  
and CasaEd in
```

```
( select CasaEd  
  from DatiLibro  
  group by CasaEd  
  having count(*) >= 100 )
```

*Qualcuno potrebbe obiettare: forse la biblioteca «ha» effettivamente un libro solo se ne ha una copia, non basta che tale libro sia descritto in DatiLibro...*

6) Quando è stato pubblicato il libro **giallo** più vecchio (isbn) che sia **di una** casa editrice di cui la biblioteca ha almeno 100 diversi libri (isbn) ?

```
select min( AnnoPub )  
from Datilibro  
where Genere = "Giallo"  
and CasaEd in
```

```
( select CasaEd  
  from DatiLibro natural join Copia  
  group by CasaEd  
  having count(distinct Isbn) >= 100 )
```

*Questa versione interpreta  
più «letteralmente» la  
richiesta, verificando la  
presenza di una copia*

6) Quando è stato pubblicato il libro **giallo** più vecchio (isbn) che sia **di una** casa editrice di cui la biblioteca ha almeno 100 diversi libri (isbn) ?

```
create view CaseEdBenRapp( CE, NL ) as
( select CasaEd, count(*)
  from DatiLibro D join Copia C on C.Isbn = D.Isbn
 group by CasaEd
 having count(distinct Isbn) >= 100 )
```

```
select min(AnnoPub)
from Datilibro
where Genere = "Giallo"
   and CasaEd in
      ( select CE
        from CaseEdBenRapp )
```

```
select min(AnnoPub)
from Datilibro
      join CaseEdBenRapp
      on CasaEd = CE
where Genere = "Giallo"
```

7) Quando è stato pubblicato il libro **giallo** più vecchio (isbn) che sia **di una** casa editrice di cui la biblioteca ha almeno 100 diversi libri **gialli** (isbn) ?

```
select min(AnnoPub)
from DatiLibro
where Genere = "Giallo"
      and CasaEd in
      ( select CasaEd
        from DatiLibro natural join Copia
        where Genere = "Giallo"
        group by CasaEd
        having count(distinct Isbn) >= 100 )
```

10a) Di ogni **libro**, quanti prestiti? (*coll/isbn/titolo*)

*Usando un outer join*

```
select C.Collocazione, count( ALL CodUtente )  
from Prestito P right join Copia C on P.Coll. = C.Coll.  
group by C.Collocazione
```

*In questo modo si crea un gruppo anche per ogni Copia mai prestata, in cui gli attributi di Prestito hanno valore nullo (in virtù del right join).*

*Le tuple che formano tali gruppi, che sarebbero contate da un count(\*), non sono però contate quando si indica come argomento di count() un attributo specifico, in virtù del qualificatore ALL (che è il default, e può essere omissso).*

*Ovviamente occorre estrarre C.Collocazione, se si estraesse P.Collocazione, in corrispondenza dei conteggi pari a 0 si otterrebbe un NULL*

10a) Di ogni **libro**, quanti prestiti? (*coll/isbn/titolo*) [...e i libri **mai** presi in prestito?]

```
select Collocazione, count(*)  
from Prestito  
group by Collocazione  
union
```

```
select Collocazione, 0  
from Copia C  
where not exists ( select *  
                  from Prestito  
                  where Collocazione = C.Collocazione )
```

10a) Di ogni **libro**, quanti prestiti? (*coll/isbn/titolo*)

```
select Collocazione, count(*)  
from Prestito  
group by Collocazione  
union  
select Collocazione, 0  
from Copia  
where Collocazione not in ( select Collocazione  
                             from Prestito )
```

***Variante senza passaggio di binding***



*11a) Il libro più prestato? (coll / isbn / titolo)*

```
create view CP ( CollLib, NumP ) as  
(  select Collocazione, count(*)  
    from Prestito  
   group by Collocazione  
      union  
  select Collocazione, 0  
  from Copia C  
 where not exists ( select * from PRESTITO  
                   where Collocazione = C.Collocazione )  
)
```

```
select CollLib, NumP  
from CP  
where NumP = ( select max( NumP )  
               from CP )
```

**SENZA VIEW ?**

*11a) Il libro più prestato? (coll / isbn / titolo) SENZA VIEW*

```
select Collocazione, count(*)  
from Prestito  
group by Collocazione  
having count(*) >= ALL ( select count(*)  
                           from Prestito  
                           group by Collocazione )
```

*11a) Il libro più prestato? (coll / isbn / titolo) SENZA VIEW*

```
select Collocazione, count(*)  
from Prestito P1  
group by Collocazione  
having count(*) > ALL ( select count(*)  
                        from Prestito P2  
                        where P1.Coll <> P2.Coll  
                        group by Collocazione )
```

*Differenza rispetto alla versione precedente?*

*Questa versione estrae il massimo solo se è unico*

*In caso di "parimerito" tra due o più copie, infatti, restituisce risultato vuoto*

*12a) Il libro più prestato di ogni genere? (coll / isbn / titolo)*

```
select Genere, C.Collocazione, count(*)
from Prestito P join Copia C on P.Coll = C.Coll
      join DatiLibro D on C.Isbn = D.Isbn
group by C.Collocazione, Genere
having count(*) >= ALL (
      select count(*)
      from Prestito P2 join Copia C2 on P2.Coll = C2.Coll
            join DatiLibro on C2.Isbn = D2.Isbn
      where D2.Genere = D.Genere
      group by C2.Collocazione )
```

12a) Il libro più prestato *di ogni genere?* (coll / isbn / titolo)

```
create view QuantiPrestiti( Col, NumP, Gen ) as
select C.Collocazione, count(*), Genere
from ( Prestito P join Copia C on P.Coll = C.Coll )
      join DatiLibro D on C.Isbn = D.Isbn
group by C.Collocazione, Genere
```

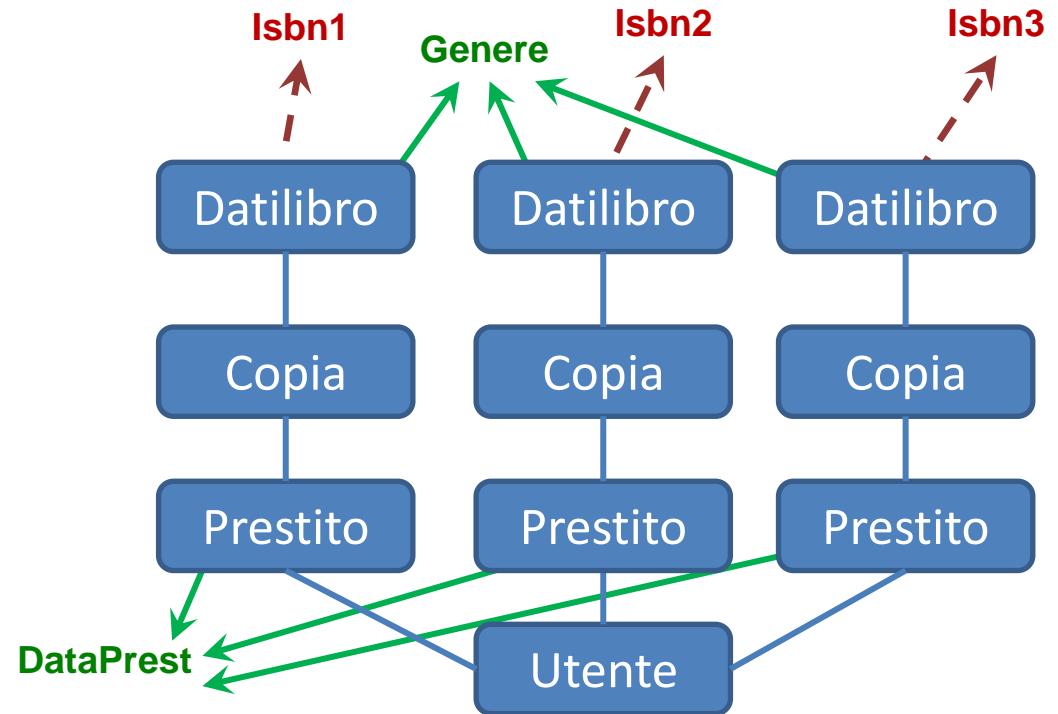
```
select *
from QuantiPrestiti Q
where NumP = ( select max(NumP)
               from QuantiPrestiti
               where Gen = Q.Gen )
```

*Le «varianti» 12b)  
12c) 13b) e 13c)  
-- dove in (b), con  
libro si intende 'isbn'  
e in (c), con libro si  
intende 'titolo'*

UTENTE(Codice, Nome, Cogn, Indiriz, Telef)    PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
 COPIA(Colloc, ISBN, DataAcquisiz)    DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

17) Utenti che hanno preso in prestito nello stesso giorno tre o più libri **diversi**, ma tutti dello stesso genere

( i libri sono **diversi** se hanno *isbn* diverso” )



UTENTE(Codice, Nome, Cogn, Indiriz, Telef)    PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
 COPIA(Colloc, ISBN, DataAcquisiz)    DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

*17) Utenti che hanno preso in prestito nello stesso giorno tre o più libri diversi,  
 ma tutti dello stesso genere*

$(\Pi_{\text{Nom Cog Cod}} \text{ UT}) \bowtie_{\text{CodUt=CodUt}}$

$\sigma_{I1 \neq I2 \wedge I2 \neq I3 \wedge I1 \neq I3} [ \rho_{I1 \leftarrow \text{Isbn}} (\text{PR} \bowtie \text{CO} \bowtie \text{DA})$

$\bowtie \text{DataPr=DataPr} \wedge \text{Genere=Genere} \wedge \text{CodUt=CodUt}$

$\rho_{I2 \leftarrow \text{Isbn}} (\text{PR} \bowtie \text{CO} \bowtie \text{DA})$

$\bowtie \text{DataPr=DataPr} \wedge \text{Genere=Genere} \wedge \text{CodUt=CodUt}$

$\rho_{I3 \leftarrow \text{Isbn}} (\text{PR} \bowtie \text{CO} \bowtie \text{DA}) ]$

UTENTE(Codice, Nome, Cogn, Indiriz, Telef)   PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
COPIA(Colloc, ISBN, DataAcquisiz)   DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

*17) Utenti che hanno preso in prestito nello stesso giorno **TRE O PIÙ** libri  
diversi, ma tutti dello stesso genere*

## IN SQL ??

```
select Nome, Cogn, Codice
from Utente
where Codice in (
    select CodUtente
    from (Prestito P join Copia C on P.Colloc = C.Colloc)
        join DatiLibro D on D.ISBN = C.ISBN
    group by CodUtente, Genere, DataPrestito
    having count(distinct D.isbn) >= 3 )
```



UTENTE(Codice, Nome, Cogn, Indiriz, Telef)   PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
COPIA(Colloc, ISBN, DataAcquisiz)   DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

*18) Utenti che hanno preso in prestito nello stesso giorno **ESATTAMENTE TRE**  
libri diversi, ma tutti dello stesso genere*

```
select Nome, Cogn, Codice
from Utente
where Codice in (
    select CodUtente
    from Prestito P join Copia C on P.Colloc = C.Colloc
    join DatiLibro D on D.ISBN = C.ISBN
    group by CodUtente, Genere, DataPrestito
    having count(distinct D.isbn) = 3 )
```

25

**E IN ALGEBRA ?? !!**

Dovrei prendere Utenti con  $\geq 3$  Prestiti e poi sottrarre quelli con  $\geq 4$

UTENTE(Codice, Nome, Cogn, Indiriz, Telef)   PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
COPIA(Colloc, ISBN, DataAcquisiz)   DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

19) *Utenti che hanno «letto» (preso in prestito!) tutti i libri*

```
select Codice, Nome, Cognome
from Utente U
where not exists (
    select *
    from Copia C
    where not exists (
        select *
        from Prestito
        where Collocazione = C.Collocazione
        and CodUtente = U.Codice ) )
```

*tutte le copie !*



UTENTE(Codice, Nome, Cogn, Indiriz, Telef)   PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
COPIA(Colloc, ISBN, DataAcquisiz)   DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

*19) Utenti che hanno «letto» (preso in prestito!) tutti i libri (coll/isbn)*

```
select Codice, Nome, Cognome
from Utente join Prestito on Codice = CodUtente
group by Codice, Nome, Cognome
having count( distinct Collocazione ) = ( select count(*)
   from Copia )
```

*19b) Utenti che hanno «letto» (preso in prestito!) tutti i libri (coll/isbn)*

```
select Codice, Nome, Cognome
from Utente join Prestito on Codice=CodUtente join Copia using (isbn)
group by Codice, Nome, Cognome
having count( distinct Isbn ) = ( select count( distinct Isbn )
                                   from Copia )
```

UTENTE(Codice, Nome, Cogn, Indiriz, Telef)    PRESTITO(Colloc, CodUtente, DataPrest, DataResa)  
COPIA(Colloc, ISBN, DataAcquisiz)    DATILIBRO(ISBN, Titolo, AnnoPub, CasaEd, PrimoAut, Genere)

```
select Codice, Nome, Cognome  
from Utente U
```

*20) Utenti che hanno preso in  
prestito libri di tutti i generi*

```
where not exists (
```

```
    select *
```

```
    from DatiLibro D
```

**Molto simile alla soluzione in TRC**

```
    where not exists (
```

```
        select *
```

```
        from Prestito P join Copia C on P.Coll = C.Coll
```

```
            join DatiLibro D1 on C.Isbn = D1.Isbn
```

```
        where CodUtente = U.Codice and
```

```
            Genere = D.Genere ) )
```

*In alternativa, si chiede che il genere non sia tra i generi di libri presi in prestito dall'utente (riducendo il numero di binding)*

```
select Codice, Nome, Cognome
from Utente U
where not exists (
  select *
  from DatiLibro
  where Genere not in (
    select Genere
    from Prestito P join Copia C on P.Coll = C.Coll
    join DatiLibro D1 on C.Isbn = D1.Isbn
    where CodUtente = U.Codice ) )
```

*Che si può scrivere anche come...*

```
select Codice, Nome, Cognome
from Utente U
where not exists (
    select *
    from DatiLibro
    where (Genere, U.Codice) not in (
        select Genere, CodUtente
        from Prestito P join Copia C on P.Coll = C.Coll
        join DatiLibro D1 on C.Isbn = D1.Isbn ) )
```

Ma il modo più «tipico» ed efficace è *sfruttare il raggruppamento e i conteggi* :

```
select Codice, Nome, Cognome
from Utente
      join Prestito P on Codice = CodUtente
      join Copia C on P.Coll = C.Coll
      join DatiLibro D on C.Isbn = D.Isbn
group by Codice, Nome, Cognome
having count( distinct Genere )
      =
      ( select count( distinct Genere )
        from Datilibro )
```

28) *L'autore più popolare tra gli utenti che hanno un altro utente loro omonimo*

```
create view UtentiConOmonimi( CodUtente, N, C, I, T ) as
  select * from Utente
  where (Nome,Cognome) in ( select Nome, Cognome
                           from Utente
                           group by Nome, Cognome
                           having count(*) > 1 )

select PrimoAutore, count(*)
from UtentiConOmonimi natural join Prestito
      natural join Copia natural join DatiLibro
group by PrimoAutore
having count(*) >= ALL (
      select count(*)
      from UtentiConOmonimi natural join Prestito
            natural join Copia natural join DatiLibro
      group by PrimoAutore )
```