



Laboratorio d'Informatica

Corso di Laurea Magistrale in Ingegneria per
l'Ambiente e il Territorio

A.A. 2021/2022

Docente: Lorenzo Putzu

Lezione 13

Basi Di Dati

E' vietata la copia, la rielaborazione, la riproduzione in qualsiasi forma dei contenuti e immagini presenti nelle lezioni. E' inoltre vietata la diffusione, la redistribuzione e la pubblicazione dei contenuti e immagini, incluse le registrazioni delle videolezioni con qualsiasi modalità e mezzo non autorizzate espressamente dall'autore o da Unica

Una o più interrogazioni?

- Esempio: trovare le matricole di tutti gli studenti e i codici di tutti gli insegnamenti. Interpretandola come una **singola** interrogazione, si potrebbe scrivere:

```
SELECT S."Matricola", I."Codice"  
FROM "Studenti" S, "Insegnamenti" I;
```

- Ma il risultato non è quello desiderato. Si tratta in realtà di due interrogazioni **indipendenti**:

- Trovare le matricole di tutti gli studenti:

```
SELECT "Matricola" FROM "Studenti";
```

- Trovare i codici di tutti gli insegnamenti:

```
SELECT "Codice" FROM "Insegnamenti";
```

Operatori insiemistici

- Dato che:
 - ogni tabella è un **insieme** di righe
 - il risultato di un'interrogazione è una **tabella**
- segue che:
 - è possibile combinare in un'unica tabella i risultati di più interrogazioni, usando gli operatori insiemistici di **unione**, **intersezione** e **differenza**, purché **tutte** le interrogazioni producano tabelle con schemi **identici** (nome, dominio e ordine delle colonne)
- Nota: le eventuali righe duplicate vengono eliminate automaticamente (non è necessario aggiungere la clausola DISTINCT).

Operatori insiemistici

- Esempio: trovare i codici dei velisti che hanno prenotato qualche barca rossa, **oppure** qualche barca verde. Una soluzione già nota:

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND  
      (B."colore" = 'rosso' OR  
       B."colore" = 'verde');
```

- Una soluzione alternativa: si potrebbero estrarre **separatamente** (con due interrogazioni **distinte**) i codici dei velisti che hanno prenotato qualche barca rossa, poi i codici dei velisti che hanno prenotato qualche barca verde, e infine eseguire l'**unione** tra i due **insiemi** di codici.

Operatori insiemistici

- Riprendendo l'esempio precedente, i codici dei velisti che hanno prenotato qualche barca rossa **oppure** qualche barca verde si possono ottenere come **unione** dei codici dei velisti che hanno prenotato qualche barca rossa, e di quelli che hanno prenotato qualche barca verde:

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'rosso'  
UNION
```

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'verde';
```

- Nota: la prima interrogazione non deve terminare con ;

Operatori insiemistici

- Esempio: trovare i codici dei velisti che hanno prenotato qualche barca rossa e qualche barca verde. Senza usare operatori insiemistici, si può essere tentati di scrivere:

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND  
      (B."colore" = 'rosso' AND  
       B."colore" = 'verde');
```

- Ma il significato di questa interrogazione non è quello desiderato; è invece: "trovare i codici dei velisti che hanno prenotato qualche barca che è **sia rossa che verde**" (e quindi il risultato sarà una tabella vuota).

Operatori insiemistici

- Nell'esempio precedente, una soluzione corretta si ottiene tramite l'**intersezione** tra i codici dei velisti che hanno prenotato qualche barca rossa, e i codici dei velisti che hanno prenotato qualche barca verde:

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'rosso'  
INTERSECT  
  
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'verde';
```

Operatori insiemistici

- Esempio: trovare i codici dei velisti che hanno prenotato qualche barca rossa, **ma non** barche verdi
- Una possibile soluzione: **sottrarre** dall'insieme dei codici dei velisti che hanno prenotato qualche barca rossa i codici di quelli che hanno prenotato qualche barca verde:

```
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'rosso'  
EXCEPT  
  
SELECT P."vid"  
FROM "Barche" B, "Prenotazioni" P  
WHERE P."bid" = B."bid" AND B."colore" = 'verde';
```

Esercizi

- Scrivere le seguenti interrogazioni con una singola istruzione SELECT, senza usare operatori insiemistici:
 - 1) Trovare nome e codice di tutti i velisti che abbiano esperienza pari a 10 o che abbiano prenotato la barca 104
 - 2) Trovare cognome e matricola di tutti gli studenti nati prima del 1/1/1990, o che abbiano sostenuto l'esame avente codice AMI01

Esercizi

- Scrivere le seguenti interrogazioni usando operatori insiemistici:
 - 3) Trovare nome e codice di tutti i velisti che abbiano esperienza pari a 10 o che abbiano prenotato la barca 104
 - 4) Trovare cognome e matricola di tutti gli studenti nati prima del 1/1/1990, o che abbiano sostenuto l'esame avente codice AMI01
 - 5) Trovare i codici delle barche che siano state prenotate da qualche velista con esperienza pari almeno a 8, e da nessun velista con età maggiore di 35 anni
 - 6) Trovare i codici degli insegnamenti i cui esami siano stati sostenuti da qualche studente nato prima del 1/1/1990, e da nessuno studente nato prima del 1/1/1985

Soluzioni

```
1) SELECT DISTINCT V."vnome", V."vid"  
   FROM "Velisti" V, "Prenotazioni" P  
   WHERE V."esperienza" = 10 OR  
         (P."vid" = V."vid" AND P."bid" = 104);
```

```
1) SELECT DISTINCT S."Cognome", S."Matricola"  
   FROM "Studenti" S, "Esami" E  
   WHERE S."Data di nascita" < '1990-01-01' OR  
         (E."Studente" = S."Matricola" AND  
         E."Esame" = 'AMI01');
```

Soluzioni

```
3) SELECT  "vnome", "vid" FROM "Velisti"
WHERE "esperienza" = 10
UNION
SELECT V."vnome", V."vid"
FROM "Velisti" V, "Prenotazioni" P
WHERE P."vid" = V."vid" AND P."bid" = 104;

4) SELECT "Cognome", "Matricola"
FROM "Studenti"
WHERE "Data di nascita" < '1990-01-01'
UNION
SELECT S."Cognome", S."Matricola"
FROM "Studenti" S, "Esami" E
WHERE E."Studente" = S."Matricola" AND
      E."Esame" = 'AMI01';
```

Soluzioni

```
5) SELECT P."bid"  
   FROM "Prenotazioni" P, "Velisti" V  
   WHERE P."vid" = V."vid" AND  
         V."esperienza" >= 8  
  
   EXCEPT  
   SELECT P."bid"  
   FROM "Prenotazioni" P, "Velisti" V  
   WHERE P."vid" = V."vid" AND V."età" > 35;
```

```
6) SELECT E."Esame"  
   FROM "Esami" E, "Studenti" S  
   WHERE S."Matricola" = E."Studente" AND  
         S."Data di nascita" < '1990-01-01'  
  
   EXCEPT  
   SELECT E."Esame"  
   FROM "Esami" E, "Studenti" S  
   WHERE S."Matricola" = E."Studente" AND  
         S."Data di nascita" < '1985-01-01';
```

Operatori insiemistici e interrogazioni nidificate

- Nella clausola WHERE di un'interrogazione Q è possibile esprimere la condizione che i valori di una o più colonne c_1, \dots, c_n debbano essere uguali a quelli di **qualche riga** (o diversi da quelli di **tutte le righe**) di una tabella T' ottenuta come risultato di un'altra interrogazione Q', purché T' contenga n colonne con gli stessi domini di c_1, \dots, c_n (nello stesso ordine):
 - SELECT ... FROM ...
WHERE ... c_1, \dots, c_n **IN** (Q') ;
 - SELECT ... FROM ...
WHERE ... c_1, \dots, c_n **NOT IN** (Q') ;

Operatori insiemistici e interrogazioni nidificate

- Esempio: Trovare i nomi dei velisti che hanno prenotato la barca n. 103. Una soluzione già nota:

```
SELECT V."vnome"  
FROM "Velisti" V, "Prenotazioni" P  
WHERE V."vid" = P."vid" AND P."bid" = 103;
```

- Una soluzione alternativa: selezionare "vnome" da "Velisti", se il valore di "vid" **appartiene** all'insieme dei codici dei velisti che hanno prenotato la barca 103:

```
SELECT "vnome" FROM "Velisti"  
WHERE "vid" IN (SELECT "vid"  
                FROM "Prenotazioni"  
                WHERE "bid" = 103);
```

Operatori insiemistici e interrogazioni nidificate

- Esempio: trovare i nomi dei velisti che **non** hanno prenotato la barca n. 103. Senza usare IN/NOT IN si può essere tentati di scrivere:

```
SELECT V."vnome"  
FROM "Velisti" V, "Prenotazioni" P  
WHERE V."vid" = P."vid" AND P."bid" <> 103;
```

- Ma il significato reale di questa interrogazione SQL è "trovare i nomi dei velisti che hanno prenotato **anche** barche diverse dalla n. 103". La soluzione corretta si può ottenere con l'operatore NOT IN:

```
SELECT "vnome" FROM "Velisti"  
WHERE "vid" NOT IN (SELECT "vid"  
FROM "Prenotazioni"  
WHERE "bid" = 103);
```

Esercizi

- 1) Trovare i cognomi degli studenti che hanno sostenuto l'esame avente codice AMI01
- 2) Trovare i nomi degli esami in cui qualche studente ha conseguito la votazione di 30

Soluzioni

```
1) SELECT "Cognome"  
FROM "Studenti"  
WHERE "Matricola" IN  
      (SELECT "Studente" FROM "Esami"  
        WHERE "Esame" = 'AMI01');
```

```
2) SELECT "Nome"  
FROM "Insegnamenti"  
WHERE "Codice" IN  
      (SELECT "Esame" FROM "Esami"  
        WHERE "Voto" = 30);
```

Operatori insiemistici e interrogazioni nidificate

- Nella clausola WHERE di un'interrogazione Q è possibile esprimere la condizione che il valore di una colonna debba essere maggiore, minore, ecc. di **qualcuno** dei valori, o di **tutti** i valori, contenuti in una tabella ottenuta come risultato di un'altra interrogazione Q', avente una sola colonna **con lo stesso dominio**:
 - SELECT ... FROM ...
WHERE ... "colonna" operatore-confronto **ANY** (Q') ;
 - SELECT ... FROM ...
WHERE ... "colonna" operatore-confronto **ALL** (Q') ;
- dove **operatore-confronto** è uno degli operatori già noti:
 - < <= = => = <>

Esempio

- Trovare le matricole degli studenti che hanno ottenuto il voto più alto nell'esame AMI01 (cioè, un voto maggiore o uguale a quello di **tutti** gli altri esami dello stesso insegnamento):

```
SELECT "Studente" FROM "Esami"  
WHERE "Esame"='AMI01' AND "Voto" >= ALL  
      (SELECT "Voto" FROM "Esami"  
       WHERE "Esame"='AMI01');
```

Esempio

- Trovare tutti i dati sui velisti la cui esperienza è maggiore di quella di **qualche** velista di nome Lubber:

```
SELECT *
FROM "Velisti"
WHERE "esperienza" > ANY
    (SELECT "esperienza"
     FROM "Velisti"
     WHERE "vnome" = 'Lubber');
```

Esercizi

- Scrivere le seguenti interrogazioni usando gli operatori ANY e ALL:
 - 1) Trovare i codici dei velisti più esperti
 - 2) Trovare nome ed età dei velisti più anziani
 - 3) Trovare i cognomi degli studenti più giovani
 - 4) Trovare le matricole degli studenti che non hanno ottenuto il voto più basso nell'esame AMI01 (cioè, che hanno ottenuto un voto maggiore rispetto a **qualche** altro studente)

Soluzioni

- 1)

```
SELECT "vid" FROM "Velisti"  
WHERE "esperienza" >= ALL  
      (SELECT "esperienza" FROM "Velisti");
```
- 2)

```
SELECT "vnome", "età" FROM "Velisti"  
WHERE "età" >= ALL  
      (SELECT "età" FROM "Velisti");
```
- 3)

```
SELECT "Cognome" FROM "Studenti"  
WHERE "Data di nascita" >= ALL  
      (SELECT "Data di nascita" FROM "Studenti");
```
- 4)

```
SELECT "Studente" FROM "Esami"  
WHERE "Esame"='AMI01' AND E."Voto" > ANY  
      (SELECT "Voto" FROM "Esami"  
       WHERE "Esame" = 'AMI01');
```

Operatori di aggregazione

- Consentono di esprimere interrogazioni che calcolano un valore **aggregato**:
 - il numero di righe di una tabella che soddisfano una data condizione; esempi:
 - trovare il numero di soci del circolo nautico
 - trovare il numero dei soci aventi 35 anni
 - la somma, la media, il minimo o il massimo dei valori di una colonna;
esempio:
trovare l'età media di tutti i velisti
- Restituiscono **un numero**, contenuto in una tabella con una sola colonna e una sola riga

Operatori di aggregazione

- Nella clausola SELECT:
 - COUNT (*): conta le righe selezionate dall'interrogazione (se sono coinvolte più tabelle, s'intendono le righe selezionate nel loro prodotto cartesiano)
 - COUNT (DISTINCT "colonna", ...): conta le righe contenenti valori **distinti** delle colonne indicate
 - SUM ([DISTINCT] "colonna"): calcola la somma dei valori di **colonna** nella righe selezionate (se è presente DISTINCT, i valori duplicati vengono contati una sola volta)
 - AVG ([DISTINCT] "colonna"): come sopra (valor medio)
 - MAX ("colonna"): come sopra (valore più grande)
 - MIN ("colonna"): come sopra (valore più piccolo)

Esempi

- Trovare il numero dei velisti soci del circolo nautico:

```
SELECT COUNT (*) FROM "Velisti";
```
- Trovare l'età media di tutti i velisti:

```
SELECT AVG ("età") FROM "Velisti";
```
- Trovare il numero di valori distinti di esperienza dei velisti:

```
SELECT COUNT (DISTINCT "esperienza")  
FROM "Velisti";
```
- Trovare l'età media dei velisti con esperienza 10:

```
SELECT AVG ("età") FROM "Velisti"  
WHERE "esperienza" = 10;
```

Operatori di aggregazione e interrogazioni nidificate

- Esempio: trovare nome ed età dei velisti più anziani (cioè di quelli con età pari all'età **più grande** tra tutti i velisti).
- Una soluzione con l'operatore ALL:

```
SELECT "vnome", "eta" FROM "Velisti"  
WHERE "eta" >= ALL (SELECT "eta" FROM "Velisti");
```

- Una soluzione alternativa con un operatore di aggregazione:

```
SELECT "vnome", "eta" FROM "Velisti"  
WHERE "eta" = (SELECT MAX ("eta")  
FROM "Velisti");
```

Esercizi

- Scrivere le seguenti interrogazioni usando operatori di aggregazione:
 - 1) Trovare i cognomi degli studenti più giovani
 - 2) Trovare i nomi dei velisti più esperti
 - 3) Trovare le matricole degli studenti che non hanno ottenuto il voto più basso nell'esame AMI01

Soluzioni

- 1)

```
SELECT "Cognome" FROM "Studenti"  
WHERE "Data di nascita" =  
      (SELECT MAX ("Data di nascita")  
       FROM "Studenti");
```
- 2)

```
SELECT "vnome" FROM "Velisti"  
WHERE "esperienza" =  
      (SELECT MAX ("esperienza")  
       FROM "Velisti");
```
- 3)

```
SELECT "Studente" FROM "Esami"  
WHERE "Esame" = 'AMI01' AND "Voto" >  
      (SELECT MIN ("Voto") FROM "Esami"  
       WHERE "Esame" = 'AMI01');
```