

# Esercitazione 4

Intelligenza Artificiale a.a. 2021-22

C.D. Hromei (Università di Roma Tor Vergata)

# Obiettivi

---

1. Vedere le differenze tra una query in **SQL** e **FOL**
2. Trasformare una formula in un insieme di clausole
3. Vedere un esempio di risoluzione per refutazione

# Un database musicale

ENTITA'

ATTRIBUTO

VINCOLO

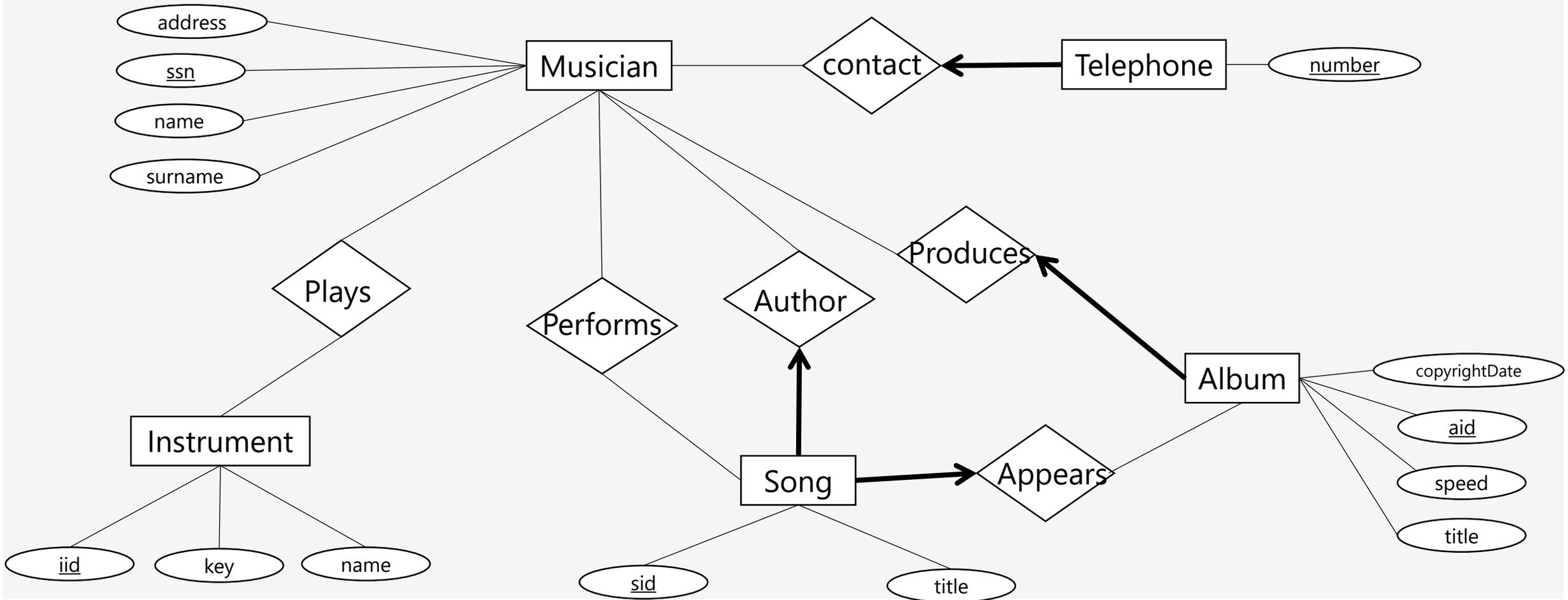
RELAZIONE

Una casa discografica decide di organizzare in un archivio i dati sui musicisti che suonano nei dischi di sua produzione.

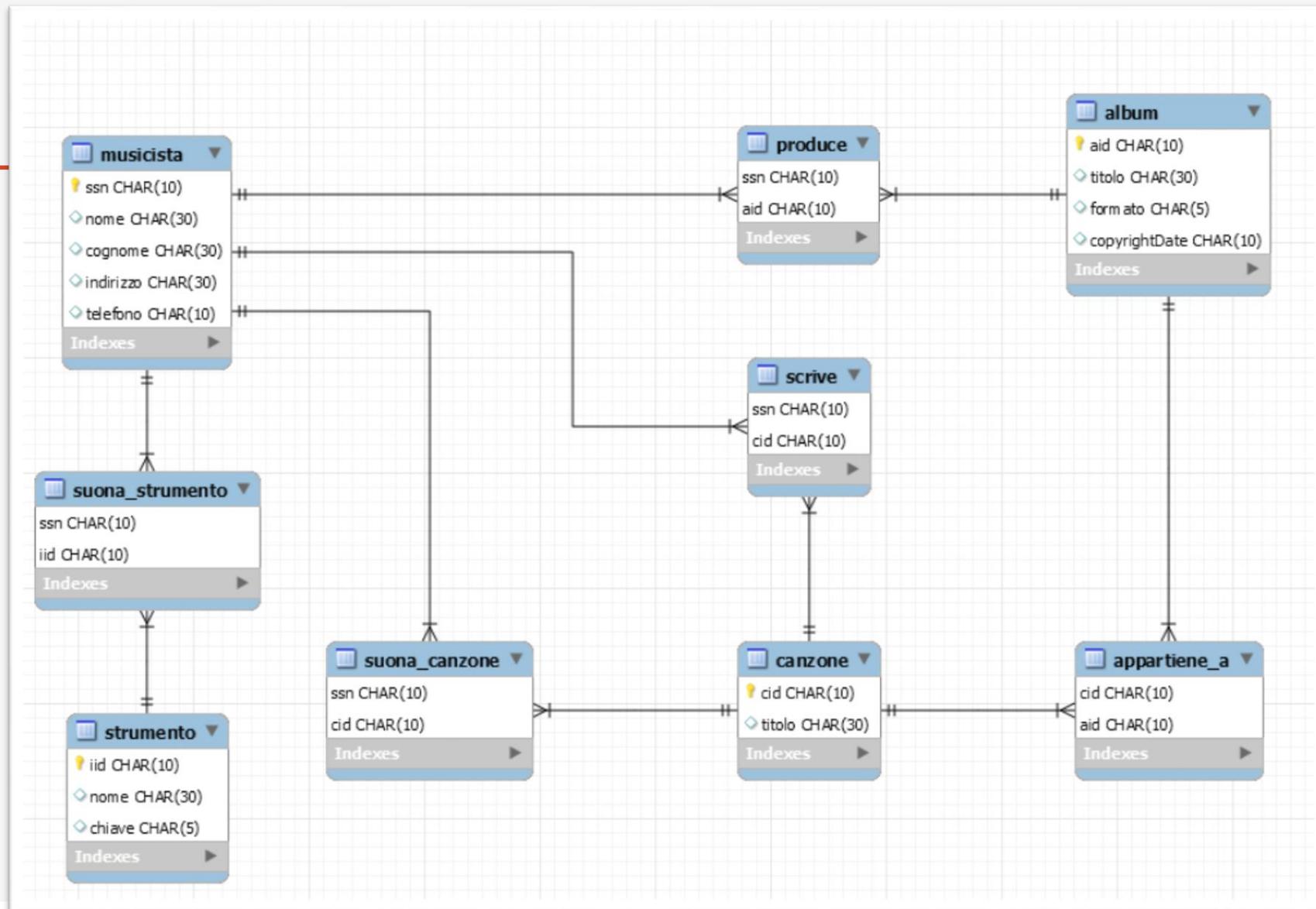
SPECIFICHE:

- Ogni **musicista** deve essere registrato con i suoi dati anagrafici (**nome**, **cognome**, **indirizzo** e **ssn**). Allo scopo di rintracciare i **musicisti** all' occorrenza, **devono essere rappresentati** i loro (anche **molteplici**) **numeri telefonici**. Gli **strumenti suonati** da ciascun **musicista** vanno registrati con le loro **chiavi**."
- Ogni **brano ha un solo autore**. Per ogni **album** è importante registrare la **data di deposito** (copyright date) e il **formato** (come ad es. CD o MC). Ogni **album ha esattamente un musicista** che svolge il ruolo di **produttore**. **Nessuna canzone** può **apparire in più di un album**, una **canzone** può **essere suonata da uno o più musicisti**.

# Schema ER



# Schema fisico



# Una knowledge base musicale

---

```
// musician(SSN, Name, Telephone, CityAddress).  
musician(1, johnLennon, 338545566, liverpool).  
musician(2, paulMcCartney, 3254566, liverpool).
```

```
// instrument(IID, Name, Key).  
instrument(1, bass_guitar, g).  
instrument(2, lead_vocal, c).  
instrument(3, keyboards, c).  
// plays(SSN, InstrumentID)
```

...

```
// performs(SSN, SongID)
```

...

```
// song_appears(SongID, Author, Title, AlbumID).
```

```
song_appears(1, paulMcCartney, sgtPeppersLonelyHeartsClubBand, 2).  
song_appears(2, johnLennon, withALittleHelpFromMyFriends, 2).  
song_appears(2, paulMcCartney, withALittleHelpFromMyFriends, 2).
```

...

```
song_appears(14, georgeHarrison, taxman, 1).  
song_appears(15, johnLennon, eleanorRigby, 1).
```

...

```
song_appears(23, paulMcCartney, gotToGetYouIntoMyLife, 1).  
song_appears(25, mickJagger, brownSugar, 3).  
song_appears(26, mickJagger, sway, 3).
```

...

# Queries d'esempio

---

1. Quali canzoni ha scritto McCartney?
2. Quante canzoni ci sono nell'album Revolver?
3. Quali strumenti suonano i musicisti che hanno prodotto l'album del 1969?
4. Qual è il numero di telefono di un musicista che suona la chitarra ed è anche autore di almeno 1 brano?
5. Quali musicisti hanno partecipato nell'album Yellow submarine?

# 1. Quali canzoni ha scritto McCartney?

---

## SQL

```
SELECT canzone.titolo  
  
FROM canzone, scrive, musicista  
  
WHERE  
  
    musicista.cognome = "McCartney" AND  
  
    musicista.ssn= scrive.ssn AND  
  
    scrive.cid = canzone.cid;
```

## FOL

$\exists x$   $\alpha$

musician( $\_$ , "paulMcCartney",  $\_$ ,  $\_$ )  $\wedge$   
song\_appears( $\_$ , "paulMcCartney",  $x$ ,  $\_$ )

// song\_appears(SongID, Author, Title, AlbumID).

// musician(SSN, Name, Telephone, CityAddress).

# Trasformazione di $\alpha$ (1)

---

$\alpha = \exists x (\text{musician}(\_, \text{"paulMcCartney"}, \_, \_) \wedge \text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)).$

1. Eliminazione di implicazioni ✓
2. Portare negazioni all'interno ✓
3. Standardizzazione variabili ✓
4. Skolemizzazione

**N.B.** trattiamo  $\_$  come una variabile di cui non ci interessa l'esito né il valore, **NON** ci preoccupiamo di standardizzarla

$\text{musician}(\_, \text{"paulMcCartney"}, \_, \_) \wedge \text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)$

# Trasformazione di $\alpha$ (2)

---

- 5. Eliminazione quantificatori universali: 
- 6. Conjunctive Normal Form 
- 7. Notazione a clausole:  $\{\text{musician}(\_, \text{"paulMcCartney"}, \_, \_)\}, \{\text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)\}$
- 8. Separazione variabili 

$\alpha = \{\text{musician}(\_, \text{"paulMcCartney"}, \_, \_)\}, \{\text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)\}$

# Risoluzione per *refutazione*

---

La nostra KB  $\models \alpha$  ?

Teorema di *refutazione*:

“KB  $\cup \{\neg\alpha\}$  è insoddisfacibile sse KB  $\models \alpha$ ”

E “KB è insoddisfacibile sse KB  $\vdash_{\text{RES}} \{\}$ ”

Dobbiamo quindi aggiungere la negazione di  $\alpha$  e cercare di generare la clausola vuota!

# Risoluzione per *refutazione* di $\neg\alpha$ (1)

---

$\neg\alpha = \neg(\text{musician}(\_, \text{"paulMcCartney"}, \_, \_) \wedge \text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_))$

$= \neg\text{musician}(\_, \text{"paulMcCartney"}, \_, \_) \vee \neg\text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)$

$= \{\neg\text{musician}(\_, \text{"paulMcCartney"}, \_, \_), \neg\text{song\_appears}(\_, \text{"paulMcCartney"}, x, \_)\}$

# Risoluzione per *refutazione* di $\neg\alpha$ (2)

$\{\neg\text{musician}(\_, \text{PaulMcCartney}, \_, \_), \neg\text{song\_appears}(\_, \text{PaulMcCartney}, x, \_)\}$

$\{\text{musician}(1, \text{JohnLennon}, 338545566, \text{Liverpool})\}$

$\{\text{musician}(2, \text{PaulMcCartney}, 3254566, \text{Liverpool})\}$

$\{\text{instrument}(1, \text{Bass\_guitar}, \text{G})\}$

$\{\text{instrument}(2, \text{Lead\_vocal}, \text{C})\}$

$\{\text{instrument}(3, \text{Keyboards}, \text{C})\}$

$\{\text{song\_appears}(1, \text{PaulMcCartney}, \text{SgtPeppersLonelyHeartsClubBand}, 2)\}$

$\{\text{song\_appears}(2, \text{JohnLennon}, \text{WithALittleHelpFromMyFriends}, 2)\}$

$\{\text{song\_appears}(23, \text{PaulMcCartney}, \text{GotToGetYouIntoMyLife}, 1)\}$

$\{\text{song\_appears}(25, \text{MickJagger}, \text{BrownSugar}, 3)\}$

$\{\neg\text{song\_appears}(\_, \text{PaulMcCartney}, x, \_)\}$

$\{\}$



**Ma era l'unico modo?**

KB

# Risoluzione per *refutazione* di $\neg\alpha$ (2b)

$\{\neg\text{musician}(\_, \text{PaulMcCartney}, \_, \_), \neg\text{song\_appears}(\_, \text{PaulMcCartney}, x, \_)\}$

$\{\text{musician}(1, \text{JohnLennon}, 338545566, \text{Liverpool})\}$

$\{\text{musician}(2, \text{PaulMcCartney}, 3254566, \text{Liverpool})\}$

$\{\text{instrument}(1, \text{Bass\_guitar}, \text{G})\}$

$\{\text{instrument}(2, \text{Lead\_vocal}, \text{C})\}$

$\{\text{instrument}(3, \text{Keyboards}, \text{C})\}$

$\{\text{song\_appears}(1, \text{PaulMcCartney}, \text{SgtPeppersLonelyHeartsClubBand}, 2)\}$

$\{\text{song\_appears}(2, \text{JohnLennon}, \text{WithALittleHelpFromMyFriends}, 2)\}$

$\{\text{song\_appears}(23, \text{PaulMcCartney}, \text{GotToGetYouIntoMyLife}, 1)\}$

$\{\text{song\_appears}(25, \text{MickJagger}, \text{BrownSugar}, 3)\}$

$\{\neg\text{musician}(\_, \text{PaulMcCartney}, \_, \_)\}$

$\{\}$



**Ma era l'unico modo?**

KB

# Risoluzione per *refutazione* di $\neg\alpha$ (2c)

$\{\neg\text{musician}(\_, \text{PaulMcCartney}, \_, \_), \neg\text{song\_appears}(\_, \text{PaulMcCartney}, x, \_)\}$

$\{\text{musician}(1, \text{JohnLennon}, 338545566, \text{Liverpool})\}$

$\{\text{musician}(2, \text{PaulMcCartney}, 3254566, \text{Liverpool})\}$

$\{\text{instrument}(1, \text{Bass\_guitar}, \text{G})\}$

$\{\text{instrument}(2, \text{Lead\_vocal}, \text{C})\}$

$\{\text{instrument}(3, \text{Keyboards}, \text{C})\}$

$\{\text{song\_appears}(1, \text{PaulMcCartney}, \text{SgtPeppersLonelyHeartsClubBand}, 2)\}$

$\{\text{song\_appears}(2, \text{JohnLennon}, \text{WithALittleHelpFromMyFriends}, 2)\}$

$\{\text{song\_appears}(23, \text{PaulMcCartney}, \text{GotToGetYouIntoMyLife}, 1)\}$

$\{\text{song\_appears}(25, \text{MickJagger}, \text{BrownSugar}, 3)\}$

$\{\neg\text{musician}(\_, \text{PaulMcCartney}, \_, \_)\}$

$\{\}$



**Ma era l'unico modo?  
Penso che ormai sia chiaro.. 😊**

KB

# SummarAlzing 😊

- I database e le basi di conoscenza sono strumenti completamente diversi, ma possono rappresentare lo stesso mondo, da punti di vista diversi.
- I linguaggi logici vengono usati come metodi di formalizzazione del dominio per ricercare, attraverso la deduzione, soluzioni a problemi. I metodi di risoluzione, come quello per *refutazione*, trasformano le formule e applicano meccanismi di unificazione per ricercare soluzioni.
- Al contrario, i linguaggi come SQL richiedono che sia formalizzato il metodo con cui ricercare i dati, piuttosto che le loro proprietà.