

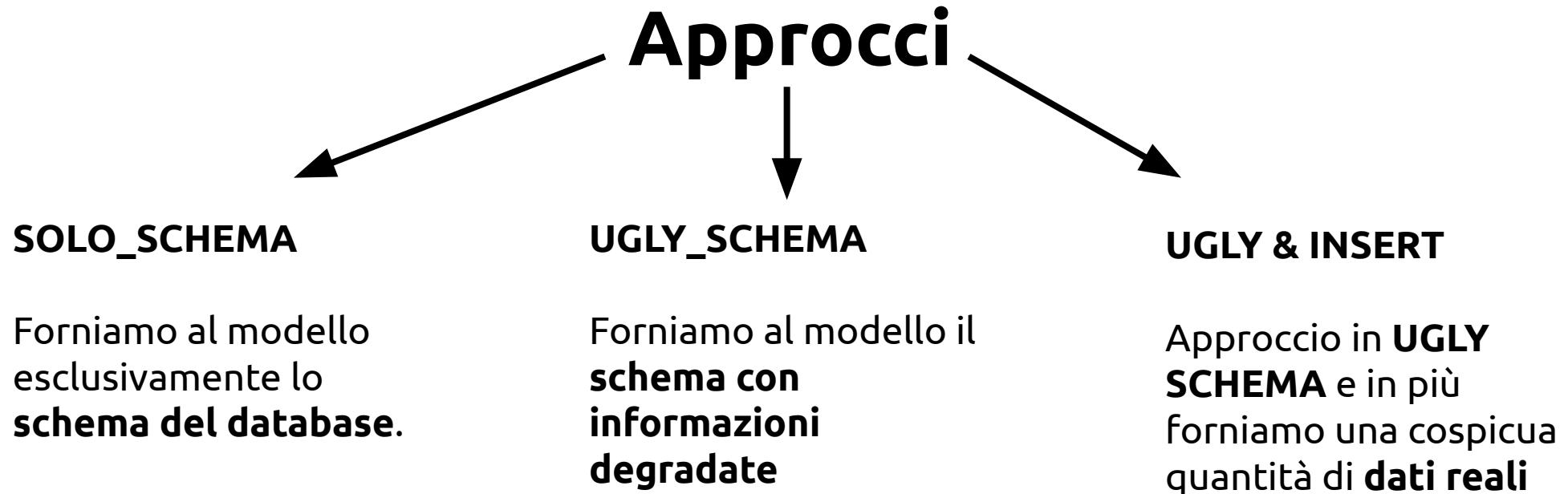
Text-To-SQL prompting LLMs

Problema: effettuare la traduzione di query in linguaggio naturale in query SQL.

Goal: testare le performance dei vari LLMs(al momento usiamo Claude di Anthropic) sul problema in questione usando database di esempio.

Cosa sa il LLM sul database?

Vorremo dare in pasto al modello il minor numero di informazioni possibili sul database.
In particolare vogliamo mantenere riservati i valori contenuti nel database che potrebbero essere
informazioni personali etc...



Testing e Valutazione dei modelli

- Scegliamo un database()
- Consideriamo uno dei 3 approcci alla volta
- Definiamo una serie di query in linguaggio naturale correlate al contenuto del database
- Attribuiamo ad ogni query un certo livello di difficoltà
- Chiediamo al modello di tradurre la query da linguaggio naturale a sql
- Verifichiamo se il modello ha tradotto le query in maniera giusta

Query

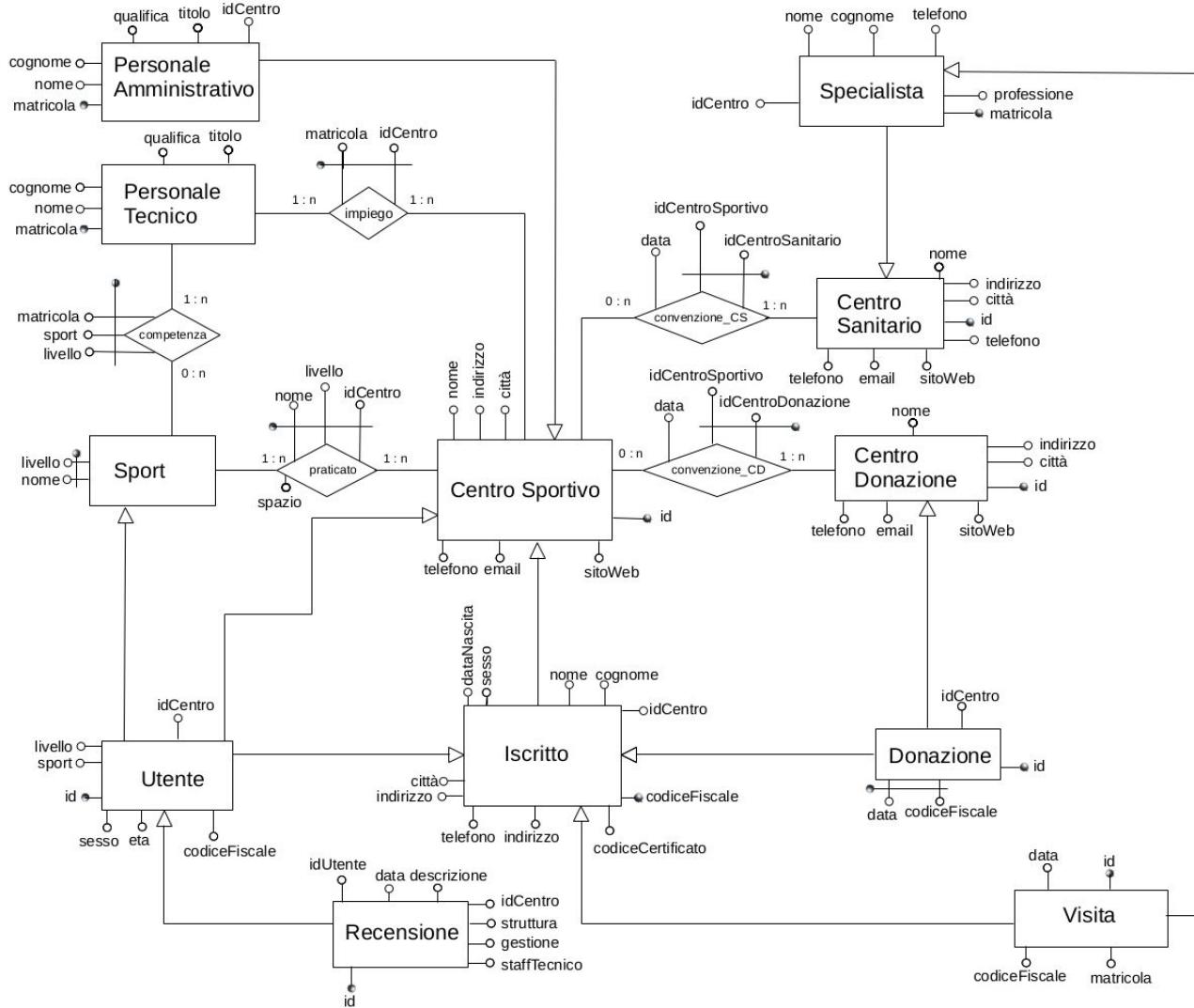
Definiamo 3 livelli di difficoltà per le varie queries (o meglio come ci aspettiamo siano le rispettive *migliori* traduzioni in sql) :

EASY: selezione diretta priva di join e funzioni di aggregazioni semplici(MIN,COUNT,AVG...)

MEDIUM: selezione con join di al massimo 2 tavelle e funzioni di aggregazione di qualsiasi tipo (group by,having...)

HARD: selezione con join di un numero arbitrario di tavelle e funzioni di aggregazione di qualsiasi tipo (group by,having...)

Schema ER-fisico



Database centri sportivi

Database Centri Sportivi

- Caratteristiche:
- Complessità medio-bassa(17 tabelle)
- Presenza di ambiguità (tabelle diverse con attributi con lo stesso nome)
- Non è presente sul Web (quindi è da escludere che il modello lo abbia processato in fase di training)
- Può contenere informazioni sensibili/riservate
(recensioni,visite,donazioni...)
-

Query 1

Livello : 1

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Seleziona nome, indirizzo, sito web dei Centri Sportivi di Roma."

Ecco la query SQL:

```
SELECT nome, indirizzo, sitoWeb  
FROM CentroSportivo  
WHERE citta = 'Roma';
```

Query 1- schema ugly

Livello : 1

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Seleziona nome, indirizzo, sito web dei Centri Sportivi di Roma."

Ecco la query SQL tradotta:

sql

Copy

```
SELECT nm, ndrzz, stWb  
FROM cntrsprtv  
WHERE ctt = 'Roma'
```

La query seleziona:

- nome (nm)
- indirizzo (ndrzz)
- sito web (stWb)

dalla tabella cntrsprtv

dove la città (ctt) è uguale a "Roma".

CLAUDE INSTANT

Query 1- ugly & insert

Livello : 1

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona nome, indirizzo, sito web dei Centri Sportivi di Roma."

Ecco la query SQL per ottenere nome, indirizzo e sito web dei Centri Sportivi di Roma:

```
sql Copy
SELECT nm, ndrzz, stWb
FROM cntrsprtV
WHERE ctt = 'Roma'
```

La tabella cntrsprtV contiene i dati sui Centri Sportivi, le colonne nm, ndrzz e stWb contengono rispettivamente:

- nm: nome del centro sportivo
- ndrzz: indirizzo
- stWb: sito web

...

La condizione WHERE ctt = 'Roma' filtra i risultati in modo da restituire solo i centri sportivi di Roma (ctt è la colonna con la città).

Quindi questa query restituirà un risultato come:

nm	ndrzz	stWb
Sport Center	via Italo 869	www.SportCenter.it
Atletigame	via dei Draghi 345	Atletigame.com

CLAUDE INSTANT

Query 2

Livello : 1

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"

Ecco la traduzione in SQL della query:

```
SELECT nome, cognome, dataNascita, email  
FROM iscritto  
WHERE sesso = 'femminile'  
LIMIT 5
```

Questa query seleziona:

...

- nome
 - cognome
 - dataNascita
 - email
- dalla tabella iscritto
dove sesso è 'femminile'
e limita i risultati a 5 righe.

Query 2 -schema ugly

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"

Ecco la query SQL tradotta:

```
sql Copy
SELECT nm, cgnm, dtNsct, ml
FROM scrtt
WHERE sss = 'femminile'
LIMIT 5
```

La query seleziona:

- Nome (nm)
- Cognome (cgnm)
- Data di nascita (dtNsct)
- Email (ml)

dalla tabella scrtt

dove il sesso (sss) è femminile

e limita il risultato a 5 righe usando LIMIT 5.

CLAUDE INSTANT

Query 2 - ugly & insert

Livello : 1

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"

Ecco la traduzione della query:

```
sql Copy
SELECT nm, cgnm, dtNsct, ml
FROM scrtt
WHERE sss = 'femminile'
LIMIT 5
```

La tabella scrtt contiene i dati sugli iscritti, con le seguenti colonne:

- nm: nome
- cgnm: cognome
- dtNsct: data di nascita
- ml: email
- sss: sesso (maschile o femminile)

...

La condizione WHERE sss = 'femminile' filtra i risultati in modo da restituire solo gli iscritti di sesso femminile.

LIMIT 5 limiterà il risultato a solo 5 righe.

Quindi il risultato potrebbe essere ad esempio:

nm	cgnm	dtNsct	ml
Sofia	Chiari	2005-11-12	sofiachiari@hotmail.it

CLAUDE INSTANT

Query 3

Livello : 1

Esito:***Corretto**

Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"

La query tradotta in SQL è:

```
SELECT nome, livello
FROM sport
JOIN praticato ON sport.nome = praticato.sport AND sport.livello =
praticato.livello
WHERE spazio = 'indoor'
```

Questa query seleziona:

- nome
 - livello
- dalla tabella sport
mediante un JOIN con la tabella praticato
sulla base delle colonne nome e livello in entrambe le tabelle
e applica la condizione WHERE spazio = 'indoor' per selezionare
solo gli sport praticati al chiuso.

In sintesi:

- Viene utilizzato un JOIN tra le tabelle sport e praticato per ottenere le informazioni sugli sport da entrambe le tabelle
- La clausola WHERE filtra i risultati in modo da ottenere solo gli sport praticati al chiuso (spazio = 'indoor')
- Vengono selezionate solo le colonne nome e livello dagli sport restituiti

...

***La query poteva essere fatta solo sulla tabella praticato senza la necessità di fare join con la tabella sport**

CLAUDE INSTANT

Query 3- schema ugly

Livello : 1

Esito:***Corretto**

Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"

Ecco la query SQL tradotta:

```
sql Copy
SELECT sprt, lvll
FROM prtct
WHERE spz = 'indoor'
```

La query seleziona:

- Nome sport (sprt)
- Livello (lvll)

...

dalla tabella prtct

dove la zona (spz) è indoor (al chiuso).

*Meglio che nella versione con schema originale. Non include tabelle inutili.

Query 3- ugly & insert

Livello : 1

Esito:***Scorretto**

Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"

Ecco la traduzione in SQL:

```
sql
SELECT nm, lvll
FROM sprt
WHERE tp = 'al chiuso'
```

La tabella sprt contiene i dati sugli sport, con le seguenti colonne:

- nm: nome dello sport
- lvll: livello (amatoriale, agonistico, professionale)
- tp: tipo (all'aperto, al chiuso)

La condizione WHERE tp = 'al chiuso' filtra i risultati in modo da restituire solo gli sport praticati al chiuso.

Quindi il risultato potrebbe contenere righe come:

...

nm	lvll
Pallavolo	agonistico
Calcetto	amatoriale
Bodybuilding	professionale
Nuoto	agonistico

*La struttura della query è corretta ma la colonna "tp"(non esistente) viene sostituita con la colonna spazio "spz". Ci aspettavamo che "al chiuso" venisse tradotto con "indoor" valore presente nel dump per la colonna "spz".

CLAUDE INSTANT

Query 4

Livello : 1,5

Esito:***Corretto**

Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"

Ecco la query tradotta in SQL:

```
SELECT nome, cognome, telefono, email, professione  
FROM specialista  
ORDER BY professione
```

Questa query seleziona:

- nome
 - cognome
 - telefono
 - email
 - professione
- dalla tabella specialista
e ordina i risultati per professione in ordine crescente come impostato da ORDER BY professione

Riassumendo:

- Viene selezionata solo la tabella specialista
- Vengono selezionate le colonne:
 - nome
 - cognome
 - telefono
 - email
 - professione

***si osservi che
“sanitario”, che si
riferisce a Specialista
non viene connesso
erroneamente con la
tabella
CentroSanitario**

CLAUDE INSTANT

Query 4- schema ugly

Livello : 1,5

Esito:***Corretto**

Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"

Ecco la query SQL tradotta:

sql

Copy

```
SELECT nm, cgnm, tlfn, ml, prfssn  
FROM spclst  
ORDER BY prfssn
```

La query seleziona:

- Nome (nm)
- Cognome (cgnm)
- Numero di telefono (tlfn)
- Email (ml)
- Professione (prfssn)

...

dalla tabella spclst (sanitari)

ordinati in modo crescente per prfssn (professione).

*si osservi che
“sanitario”,che si
riferisce a Specialista
non viene connesso
erroneamente con la
tabella
CentroSanitario

CLAUDE INSTANT

Query 4- ugly & insert

Livello : 1,5

Esito:***Scorretto**

Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"

Ecco la traduzione della query in SQL:

sql

Copy

```
SELECT nm, cgnm, cntts, prfsn  
FROM prsnltcnc  
ORDER BY prfsn
```

La tabella prsnltcnc contiene i dati sul personale sanitario, con le seguenti colonne:

- nm: nome
- cgnm: cognome
- cntts: contatti (telefono, email)
- prfsn: professione (medico, infermiere, fisioterapista...)

...

L'istruzione ORDER BY prfsn ordina i risultati in base alla professione.

Ad esempio il risultato potrebbe essere:

nm	cgnm	cntts	prfsn
Carlo	Santoro	010-123456789, carlosantoro@gmail.com	medico

***Viene confuso personalesanitario(q uindi specialista) con personeatecnico.**

CLAUDE INSTANT

Query 5

Livello : 2

Esito:**Corretto**

Traduci in sql la seguente query :"matricola, nome, cognome, ruolo del membro del personale amministrativo con diploma che ha un impiego nel centro sportivo "Appia Sport Village"

Ecco la traduzione in SQL della query:

```
SELECT p.matricola, p.nome, p.cognome, p.qualifica  
FROM personaleamministrativo p  
JOIN centrosportivo c ON p.idCentro = c.id  
WHERE p.titolo = 'diploma' AND c.nome = 'Appia Sport Village'
```

La query seleziona:

- La matricola
- Il nome
- Il cognome
- La qualifica (ruolo)

Dalla tabella personaleamministrativo p
Attraverso un JOIN con la tabella centrosportivo c sulla base
della colonna idCentro

...

E applica le condizioni:

- WHERE p.titolo = 'diploma' per selezionare solo il personale amministrativo con diploma
- WHERE c.nome = 'Appia Sport Village' per selezionare solo il personale che lavora presso quel centro sportivo

Riassumendo:

- Viene effettuato un JOIN tra personaleamministrativo e centrosportivo per collegare l'informazione sul centro sportivo

CLAUDE INSTANT

Query 5 -schema ugly

Livello : 2

Esito:***Scorretto**

Traduci in sql la seguente query :"matricola,nome,cognome,ruolo del membro del personale amministrativo con diploma che ha un impiego nel centro sportivo "Appia Sport Village"

Ecco la query SQL tradotta per rispondere alla domanda:

```
sql Copy
SELECT mtrcl, nm, cgnm, qlfc
FROM prsnlmmnstrtv
WHERE ttl = 'diploma'
AND dCntr = (
    SELECT id FROM cntrsprtv
    WHERE stWb = 'Appia Sport Village'
)
```

La query seleziona:

- Matricola (mtrcl)
- Nome (nm)
- Cognome (cgnm)
- Ruolo (qlfc)

...

*Il valore "Appia Sport Village" dovrebbe essere egualato al campo nm("nome") e non al campo stWb("sitoWeb").
Inoltre questa query poteva benissimo esser fatta senza la select annidata

CLAUDE INSTANT

Query 5 - ugly & insert

Livello : 2

Esito:***Scorretto**

Traduci in sql la seguente query :"matricola, nome, cognome, ruolo del membro del personale amministrativo con diploma che ha un impiego nel centro sportivo "Appia Sport Village"

Ecco la traduzione in SQL:

```
sql Copy
SELECT mtctrl, nm, cgnm, rl
FROM prsnlmmnstrtv
WHERE qflfcn= 'diploma' AND cntrsprtv='Appia Sport Village'
```

La tabella prsnlmmnstrtv contiene i dati sul personale amministrativo, con le colonne:

- mtctrl: matricola
- nm: nome
- cgnm: cognome
- rl: ruolo
- qflfcn: titolo di studio (laurea,diploma)
- cntrsprtv: centro sportivo di lavoro

...

La condizione WHERE:

- qflfcn= 'diploma' filtra i risultati restituendo solo il personale con diploma
- cntrsprtv='Appia Sport Village' filtra gli impiegati che lavorano nel centro sportivo "Appia Sport Village"

*Non esiste il campo ruolo("rl") e non viene fatto il join con la tabella "cntrsprtv"(Centro Sportivo)

CLAUDE INSTANT

Query 6

Livello : 2

Esito: Corretto

Traduci in sql la seguente query : "nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019"

Ecco la query tradotta in SQL:

```
SELECT i.nome, i.cognome, i.telefono  
FROM iscritto i  
JOIN donazione d ON i.codiceFiscale = d.codiceFiscale  
WHERE d.data > '2019-05-01'
```

La query seleziona:

- Il nome
- Il cognome
- Il telefono

Dalla tabella iscritto i

Attraverso un JOIN con la tabella donazione d sulla base della colonna codiceFiscale

E applica la condizione WHERE d.data > '2019-05-01' per selezionare solo le donazioni effettuate dopo il 1 maggio 2019.

...

Riassumendo:

- Viene effettuato un JOIN tra iscritto e donazione per collegare le informazioni sulla donazione alle informazioni sull'iscritto
- La clausola WHERE d.data > '2019-05-01' filtra i risultati in modo da ottenere solo gli iscritti che hanno donato dopo la data

CLAUDE INSTANT

Query 6 -schema ugly

Livello : 2

Esito: **Corretto**

Traduci in sql la seguente query : "nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019"

Ecco la query SQL tradotta per rispondere alla domanda:

```
sql Copy
SELECT s.nm, s.cgnm, s.tlfn
FROM scrtt s
JOIN dnzn d ON s.cdcFscl = d.cdcFscl
WHERE d.dt > '2019-05-01'
```

La query seleziona:

- Nome (s.nm)
- Cognome (s.cgnm)
- Telefono (s.tlfn)

dalla tabella scrtt (s)

attraverso un join con la tabella dnzn (d) basato sul campo cdcFscl
in comune

dove la data della donazione (d.dt) è successiva a 2019-05-01.

In questo modo otterremo i dettagli (nome, cognome e telefono)
degli iscritti che hanno effettuato almeno una donazione dopo il 1
maggio 2019.

CLAUDE INSTANT

Query 6 - ugly & insert

Livello : 2

Esito: *Scorretto

Traduci in sql la seguente query : "nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019"

Ecco la traduzione in SQL:

```
sql
SELECT scrtt.nm, scrtt.cgnm,scrtt.tlf
FROM scrtt
INNER JOIN cntrdnzn
ON scrtt.id = cntrdnzn.scrttid
WHERE cntrdnzn.dt > '2019-05-01'
```

La tabella scrtt contiene i dati sugli iscritti con:

- nm: nome
- cgnm: cognome
- tlf: telefono

...

La tabella cntrdnzn contiene i dati sulle donazioni effettuate, con un campo scrttid che collega la donazione al relativo iscritto.

Per ottenere i dati richiesti nella query, eseguiamo un INNER JOIN tra le due tabelle in base al campo scrttid.

Poi la condizione WHERE cntrdnzn.dt > '2019-05-01' filtra i risultati restituendo solo gli iscritti che hanno effettuato almeno una donazione dopo il 01/05/2019, in base alla data cntrdnzn.dt.

*Il join tra “CntrDnzn” e “scrtt” andava fatto sul campo codiceFiscale “cdcfsc1”.

La colonna “scrttd” in cntrdnzn viene inventata.

CLAUDE INSTANT

Query 7

Livello : 2,5*

Esito: Corretto

Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"

```
SELECT c.nome, AVG(r.struttura) AS media_struttura,  
AVG(r.gestione) AS media_gestione,  
AVG(r.staffTecnico) AS media_staffTecnico  
FROM CentroSportivo c  
JOIN Recensione r  
ON c.id = r.idCentro  
GROUP BY c.id
```

La query fa:

- Seleziona il nome del centro sportivo (c.nome)
- Calcola la media delle recensioni per:
 - struttura (AVG(r.struttura))
 - gestione (AVG(r.gestione))
 - staff tecnico (AVG(r.staffTecnico))

*il modello individua i 3 campi "gestione","struttura","staffTecnico" che fanno riferimento ai voti delle recensioni.

```
struttura ENUM("1", "2", "3", "4", "5")  
gestione ENUM("1", "2", "3", "4", "5")  
staffTecnico ENUM("1", "2", "3", "4", "5")  
PRIMARY KEY(c.id)
```

Query 7 -schema ugly

Livello : 2,5*

Esito: *Scorretto

Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"

Ecco la query SQL tradotta per calcolare la media delle recensioni di ogni centro sportivo:

```
sql Copy
SELECT c.nm,
       AVG(r.stffTcnc) AS media_recensioni
  FROM cntrsprtvc c
  LEFT JOIN rcnsn r
    ON c.id = r.dCntr
 GROUP BY c.id, c.nm
 ORDER BY media_recensioni DESC
```

La query:

- Seleziona il nome (c.nm) dei centri sportivi (c) dalla tabella cntrsprtvc
- Effettua un LEFT JOIN con la tabella delle recensioni (rcnsn) basato sul campo dCntr in comune
- Calcola la media (AVG) del campo r.stffTcnc (valutazione dello staff tecnico)
- Aggrega i risultati per c.id e c.nm
- Ordina i risultati per media delle recensioni in ordine discendente

In questo modo otterremo:

*il modello dovrebbe calcolare la media dei voti non solo per il campo stffTcnc(staffTecnico) ma anche gstn(gestione) e strtr(struttra).

NOTA: nello schema ugly il tipo di dato dei tre campi citati non è mascherato.
Ed è ENUM(1,2,3,4,5)

CLAUDE INSTANT

Query 7 - ugly & insert

Livello : 2,5*

Esito: ***Corretto**

Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"

```
SELECT dCntr, AVG(strttr) AS media_struttura,  
AVG(gstn) AS media_gestione,  
AVG(stfftCnc) AS media_staff_tecnico  
FROM rcnsn  
GROUP BY dCntr
```

Questa query calcola la media per:

- struttura (strttr),
 - gestione (gstn)
 - staff tecnico (stfftCnc)
- di ogni centro sportivo (gruppo per dCntr) consultando la tabella delle recensioni (rcnsn).

*Vengono calcolate 3 medie separate per i campi gstn,stfftCnc e strttr (coerente con quanto viene chiesto).

NOTA: nello schema ugly il tipo di dato dei tre campi citati non è mascherato.

Ed è ENUM(1,2,3,4,5)

Query 8

Livello : 2

Esito: Corretto

Traduci in SQL la seguente query:
"nome,cognome,centro,indirizzo,Centro,cittaCentro
,telefono degli osteopati e degli ortopedici che lavorano in un
centro
sanitario di Roma."

Here's the [SQL query](#) to retrieve the requested data:

```
SELECT nome, cognome, c.nome AS centro, indirizzo, citta, telefono
FROM specialista s
JOIN centrosanitario c ON c.id = s.idCentro
WHERE professione IN ('osteopata','ortopedico') AND citta = 'Roma'
```

This joins the `specialista` and `centrosanitario` tables, filtering the results to match:

- Rows where `professione` is either '`osteopata`' (`osteopath`) or '`ortopedico`' (`orthopedist`)
- The sanitary center (`centro`) is located in Rome, i.e. `citta` = '`Roma`'

...

The columns selected are:

- `nome` (first name)
- `cognome` (last name)
- `centro` (center name)
- `indirizzo` (address)
- `citta` (city)
- `telefono` (phone number)

*anche se non
abbiamo esplicitato,
il tipo di Centro che ci
interessa nella query,
il modello individua
Centro Sanitario
correttamente

Query 8 -schema ugly

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query:
"nome,cognome,centro,indirizzoCentro,cittaCentro
,telefono degli osteopati e degli ortopedici che lavorano in un
centro
sanitario di Roma."

Ecco la query SQL tradotta per rispondere alla domanda:

```
sql Copy
SELECT s.nm,
       s.cgnm,
       c.nm AS nome_centro,
       c.ndrzz AS indirizzo_centro,
       c.ctt AS citta_centro,
       s.tlfn
  FROM spclst s
 JOIN cntrdnzn c ON s.dCntr = c.id
 WHERE s.prfssn IN ('osteopata','ortopedico')
   AND c.ctt = 'Roma'
```

La query seleziona:

- Nome (s . nm)
- Cognome (s . cgnm)
- Nome centro (nome_centro)
- Indirizzo centro (indirizzo_centro)
- Città centro (citta_centro)
- Telefono (s . tlfn)

*Ci sono vari errori.
Il più significativo è
che non va coinvolto
CntrDnzn(CentroDon
azioni) ma
CntrSntr(CentroSanit
ario).

CLAUDE INSTANT

Query 8 - ugly & insert

Livello : 2

Esito: *Scorretto

Traduci in SQL la seguente query:

"nome,cognome,centro,indirizzoCentro,cittaCentro
telefono degli osteopati e degli ortopedici che lavorano in un
centro
sanitario di Roma."

```
SELECT nm, cgnm, dCntr, ndrzz, ctt, tlfn
FROM prsnltcnc p
INNER JOIN cntrsentr c
ON p.mtrcl = c.id
WHERE qlfc IN ('ostreopata','ortopedico')
AND ctt = 'Roma'
```

Questa query seleziona:

- nm - nome
- cgnm - cognome
- dCntr - centro sanitario in cui lavorano
- ndrzz - indirizzo del centro
- ctt - città del centro

...

*Nuovamente, viene confuso personale tecnico (prsnltcnc) con specialista(splcst). NOTA: nel dump ci sono insert per la tabella splclst che non contengono "osteopata" o "ortopedico" ma cose simili tipo "fisiatra","fisioterapista"...

CLAUDE INSTANT

Query 9

Livello : 3

Esito:**Corretta**

Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"

Ecco la query tradotta in SQL:

```
SELECT v.data, i.nome, i.cognome, i.codiceFiscale, s.professione  
FROM visita v  
JOIN iscritto i ON v.codiceFiscale = i.codiceFiscale  
JOIN specialista s ON v.matricolaSpecialista = s.matricola
```

La query seleziona:

- La data della visita (v.data)
- Il nome, cognome e codice fiscale del visitato (i.nome, i.cognome, i.codiceFiscale) dalla tabella iscritto
- La professione del visitante (s.professione) dalla tabella specialista

Attraverso JOIN:

- Tra visita e iscritto sulla base di v.codiceFiscale = i.codiceFiscale per collegare i dati del visitato
- Tra visita e specialista sulla base di v.matricolaSpecialista = s.matricola per collegare i dati del visitante

Riassumendo:

- Vengono selezionate colonne da 3 tabelle: visita, iscritto e specialista
- I JOIN tra le tabelle permettono di collegare i dati del visitato

*Visitante viene correttamente unificato con Specialista

CLAUDE INSTANT

Query 9 -schema ugly

Livello : 3

Esito:***Corretta**

Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"

Ecco la query SQL tradotta per rispondere alla domanda:

```
sql Copy
SELECT
    v.dt,
    s.nm,
    s.cgnm,
    s.cdcFscl,
    sp.prfssn
FROM vst v
JOIN spclst sp ON v.mtrclSpclst = sp.mtrcl
JOIN scrtt s ON v.cdcFscl = s.cdcFscl
```

La query seleziona:

- La data della visita (v.dt)
- Nome (s.nm)
- Cognome (s.cgnm)
- Codice fiscale (s.cdcFscl) del visitato
- Professione (sp.prfssn) del visitante

...

dalla tabella vst (visite)

attraverso due join:

1. Con spclst (specialisti) basato sul campo mtrclSpclst
Questo collega ogni visita allo specialista che l'ha effettuata
2. Con scrtt (iscritti) basato sul campo cdcFscl
Questo collega ogni visita all'iscritto visitato

In questo modo otterremo:

- La data della visita
- I dettagli (nome, cognome e codice fiscale) del visitato
- La professione del medico/specialista che ha effettuato la visita

***Visitante viene correttamente unificato con Specialista**

CLAUDE INSTANT

Query 9 - ugly & insert

Livello : 3

Esito:***Corretta**

Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"

Ecco la traduzione in SQL della query:

```
SELECT v.dt, s.nm, s.cgnm, s.cdcFscl, sp.prfssn  
FROM vst v  
JOIN scrtt s ON v.cdcFscl = s.cdcFscl  
JOIN spclst sp ON v.mtrclSpclst = sp.mtrcl
```

*Visitante viene correttamente unificato con Specialista

Le tabelle coinvolte sono:

- vst: contiene le informazioni sulle visite, con il foreign key cdcFscl che punta a scrtt (la persona visitata) e mtrclSpclst che punta a spclst (lo specialista visitante)
- scrtt: contiene le informazioni sulle persone
- spclst: contiene le informazioni sugli specialisti, compresa la

CLAUDE INSTANT

Query 10

Livello : 3.5*

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."

Ecco la query tradotta in SQL:

```
SELECT c.nome, c.città, c.telefono, s.nome AS sport
FROM centrosportivo c
JOIN praticato p ON c.id = p.idCentro
JOIN sport s ON p.sport = s.nome AND p.livello = s.livello
WHERE s.livello = 'dilettantistico'
ORDER BY sport
```

La query seleziona:

- Il nome del centro sportivo
- La città
- Il telefono
- Il nome dello sport praticato

Dalle tabelle:

- centrosportivo c
- praticato p attraverso un JOIN con c sulla base di c.id = p.idCentro
- sport s attraverso un JOIN con p sulla base di p.sport = s.nome AND p.livello = s.livello

*nella query in linguaggio naturale si chiedono i "contatti"(non sono presenti colonne con questo nome) di ogni centro sportivo e il modello restituisce una query che seleziona i valori assunti dalle colonne "indirizzo" e "email".

Query 10 - ugly schema

Livello : 3.5*

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."

```
SELECT s.nm AS "Sport",
cnt.nm AS "Centro sportivo",
cnt.tlfn AS "Telefono",
cnt.ml AS "Email"
FROM prtct
INNER JOIN cntrsprt AS cnt ON prtct.dCntr = cnt.id
INNER JOIN sprt AS s ON prtct.sprt = s.nm AND prtct.lvll = s.lvll
WHERE prtct.lvll = 'dilettantistico'
ORDER BY s.nm
```

Query 10 - ugly & insert

Livello : 3.5*

Esito: *Scorretto

Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."

Ecco la traduzione della query:

```
SELECT cntrsprt.v_nm, cntrsprt.cntt, prtct.prtct_nm  
FROM cntrsprt  
JOIN prtct ON cntrsprt.id = prtct.dCntr  
WHERE prtct.lvll = 'dilettantistico'  
ORDER BY prtct.prtct_nm
```

*Vengono inventate colonne non esistenti come "cntt" per contatti e "prtct_nm" per nome sport praticato.

Query 11

Livello : 3

Esito: *Scorretto

matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale

Here is the [SQL query](#) to retrieve the requested data:

```
SELECT p.matricola, competenza.sport, competenza.livello, c.nome  
FROM personaletecnico p  
JOIN competenza ON p.matricola = competenza.matricola  
JOIN centrosportivo c ON p.idCentro = c.id  
WHERE c.citta <> 'Roma'
```

The query does the following:

- Joins the personaletecnico, competenza and centrosportivo tables on the relevant keys
- Filters the results to only include sport centers that are NOT in Rome, i.e. WHERE c.citta <> 'Roma'
- Selects the following columns:
 - p.matricola: staff member ID
 - competenza.sport: the sport
 - competenza.livello: the level (professional or amateur)
 - c.nome: the name of the sport center

...

So this will return the [staff ID](#), sport, level and sport center name for [technical staff members](#) working in sport centers that are not in the capital (Rome).

The tables are joined as follows:

* Con “NON della capitale” ci si riferisce a Centri Sportivi che non sono nella città di Roma.
Il modello coglie correttamente questa informazione.

Tuttavia la query restituita dal modello considera per la tabella personaleTecnico il campo idCentro. Ma per la tabella in questione non esiste l’attributo idCentro.

Query 11 - ugly schema

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale"

```
SELECT m.mtrcl,  
       s.sppt,  
       s.lvll ,  
       cnt.nm AS "centro sportivo"  
  FROM prsnltcnc AS m  
 INNER JOIN cmptnz AS c ON m.mtrcl = c.mtrcl  
 INNER JOIN sppt AS s ON c.sppt = s.nm AND c.lvll = s.lvll  
 INNER JOIN mpg AS mp ON m.mtrcl = mp.mtrcl  
 INNER JOIN cntrsprt AS cnt ON mp.dCntr = cnt.id  
 WHERE cnt.ctt <> 'Roma'
```

Query 11 - ugly & insert

Livello : 3.5*

Esito: ***Corretto**

Traduci in SQL la seguente query: "matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale"

Ecco la traduzione della query:

```
SELECT mtrcl, sprt, lvll, dCntr  
FROM prsnltcnc  
JOIN mpg ON prsnltcnc.mtrcl = mpg.mtrcl  
JOIN prtct ON mpg.dCntr = prtct.dCntr  
JOIN cntrsprtv ON prtct.dCntr = cntrsprtv.id  
WHERE cntrsprtv.ctt <> 'Roma'
```

*La query passa attraverso la tabella "prtct" che ha che non è collegata con "mpg" ma con "sprt" e quindi "prsnltcnc". "NON della capitale" viene inteso correttamente. Poteva esser fatta con meno join.

Query 12

Livello : 3

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione"

```
SELECT
c.nome,
c.id,
d.nome AS nome_centro_donazione,
d.id AS id_centro_donazione,
c1.data
FROM Convenzione_CD c1
JOIN CentroSportivo c
ON c1.idCentroSportivo = c.id
JOIN CentroDonazione d
ON c1.idCentroDonazione = d.id
```

Query 12 - schema ugly

Livello : 3

Esito: **Scorretto**

...

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione"

Ecco la query SQL per ottenere i centri sportivi e di donazione con cui sono convenzionati con il nome, id e data inizio convenzione:

sql

Copy

```
SELECT c.nm, c.id, cd.dt
FROM cntrsprtvc c
JOIN (cnvnzn_cs UNION cnvnzn_cd) cd
ON c.id = cd.dCntrSprtv
JOIN (cntrsntr UNION cntrdnzn) cc
ON cd.dCntrSntr = cc.id OR cd.dCntrDnzn = cc.id
```

Query 12 - ugly & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione"

```
SELECT c.nm, c.id, cd.dt
FROM cntrsprt c
JOIN cnvnzn_cs ON c.id = cnvnzn_cs.dCntrSprtv
JOIN cntrsntr ON cntrsntr.id = cnvnzn_cs.dCntrSntr
UNION
SELECT c.nm, c.id, cd.dt
FROM cntrdnzn c
JOIN cnvnzn_cd ON c.id = cnvnzn_cd.dCntrDnzn
JOIN cntrsprt ON cntrsprt.id = cnvnzn_cd.dCntrSprtv
```

*Si effettua il join con la tabella/relazione "cnvnzn_cs" che riguarda la convenzione con i centri sanitari.

Query 1

Livello : 1

Esito: Corretto

f

traduci in mysql la query "nome,indirizzo,sitoWeb dei Centri Sportivi di Roma."



Per ottenere il nome, indirizzo e sito web dei Centri Sportivi di Roma dal database
`'centri_sportivi'`, puoi utilizzare la seguente query SQL:

sql



Copy code

```
SELECT nome, indirizzo, sitoWeb  
FROM centrosportivo  
WHERE citta = 'Roma';
```

Questo query seleziona i campi `'nome'`, `'indirizzo'` e `'sitoWeb'` dalla tabella
`'centrosportivo'` dove il campo `'citta'` è uguale a "Roma". Questo restituirà i risultati
contenenti il nome, l'indirizzo e il sito web dei centri sportivi di Roma presenti nel database.

CHAT GPT

Query 1 -schema ugly

Livello : 1

Esito: **Corretto**



Traduci in SQL la seguente query: "Seleziona nome, indirizzo, sito web dei Centri Sportivi di Roma."



La query SQL per selezionare il nome, indirizzo e sito web dei Centri Sportivi di Roma sarebbe la seguente:

sql

Copy code

```
SELECT nm AS Nome, ndrzz AS Indirizzo, stWb AS SitoWeb  
FROM cntrsprtvs  
WHERE ctt = 'Roma';
```

Questa query seleziona i campi `nm`, `ndrzz`, e `stWb` dalla tabella `cntrsprtvs`, dove il campo `ctt` (presumibilmente rappresenta la città) è uguale a 'Roma'. L'alias `AS` viene utilizzato per rinominare i nomi delle colonne nella selezione del risultato della query.

CHAT GPT

Query 1 - ugly & insert

Livello : 1

Esito: **Corretto**



Traduci in SQL la seguente query: "Seleziona nome, indirizzo, sito web dei Centri Sportivi di Roma."



Per selezionare il nome, indirizzo e sito web dei Centri Sportivi di Roma, possiamo utilizzare la seguente query SQL:



Copy code

```
sql
SELECT `nm`, `ndrzz`, `stWb`
FROM `cntrsprtv`
WHERE `ctt` = 'Roma';
```

Questa query seleziona le colonne `'nm'` (nome), `'ndrzz'` (indirizzo) e `'stWb'` (sito web) dalla tabella `'cntrsprtv'`, filtrando solo i Centri Sportivi che hanno la città (`'ctt'`) uguale a "Roma".

CHAT GPT

Query 2

Livello : 1

Esito: Corretto

f

Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"



Per selezionare il nome, cognome, data di nascita ed email di 5 iscritti di sesso femminile, puoi utilizzare la seguente query SQL:

sql Copy code

```
SELECT nome, cognome, dataNascita, email  
FROM iscritto  
WHERE sesso = 'femminile'  
LIMIT 5;
```

In questa query, stiamo selezionando i campi `nome`, `cognome`, `dataNascita` ed `email` dalla tabella `iscritto`. Utilizziamo la clausola `WHERE` per filtrare solo gli iscritti di sesso femminile. Infine, utilizziamo la clausola `LIMIT 5` per ottenere solo le prime 5 righe risultanti dalla query, che corrispondono agli iscritti femminili richiesti.

CHAT GPT

Query 2 -schema ugly

Livello : 1

Esito: ***Scorretto**



Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"



Sure! Based on the provided database model, the SQL query to select the name, surname, date of birth, and email of 5 female members would be as follows:

```
sql Copy code
SELECT nm AS nome, cgnm AS cognome, dtNsct AS "data di nascita", ml AS email
FROM tnt
WHERE sss = 'femminile'
LIMIT 5;
```

In this query, we are selecting the fields `nm` (name), `cgnm` (surname), `dtNsct` (assuming it represents the date of birth), and `ml` (email) from the table `tnt`. We use the `WHERE` clause to filter only female members, and with `LIMIT 5`, we are restricting the result to 5 records. The table `tnt` seems to contain information about the members, and the field `sss` is used to represent the gender (maschile for male and femminile for female) of the members.

*Andava coinvolta la tabella scrtt(Iscritto) e non tnt(Utente). Inoltre dtNsct(dataNsct) e ml(email) sono campi presenti solo nella tabella scrtt.

Query 2 - ugly & insert

Livello : 1

Esito: Corretto

f Traduci in SQL la seguente query: "Seleziona nome,cognome,data di nascita,email di 5 iscritti di sesso femminile"

Per selezionare il nome, cognome, data di nascita e email di 5 iscritti di sesso femminile, possiamo utilizzare la seguente query SQL:

```
sql Copy code

SELECT `nm`, `cgnm`, `dtNsct`, `ml`
FROM `scrtt`
WHERE `sss` = 'femminile'
LIMIT 5;
```

Questo query seleziona le colonne `nm` (nome), `cgnm` (cognome), `dtNsct` (data di nascita) e `ml` (email) dalla tabella `scrtt`, filtrando solo gli iscritti che hanno il sesso (`sss`) uguale a "femminile" e limitando il risultato a 5 righe.

Query 3

Livello : 1.5

Esito: *Corretto



Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"



La seguente query selezionerà il nome e il livello degli sport praticati al chiuso dalla tabella `praticato`, dove lo spazio è "indoor":

sql

Copy code

```
SELECT nome, livello  
FROM praticato  
WHERE spazio = 'indoor';
```

*Il modello coglie correttamente il fatto che con "praticati al chiuso" ci si riferisce a "indoor" valore assunto dall'attributo spazio.

CHAT GPT

Query 3 -schema ugly

Livello : 1.5

Esito: *Corretto



Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"



Based on the provided database model, the SQL query to select the name and level of sports practiced indoors would be as follows:

sql

Copy code

```
SELECT sprt.nm AS nome, sprt.lvll AS livello
FROM sprt
INNER JOIN prtct ON sprt.nm = prtct.sppt AND sprt.lvll = prtct.lvll
WHERE prtct.spz = 'indoor';
```

In this query, we are selecting the fields `nm` (name) from the `sprt` table and aliasing it as "nome", and the field `lvll` (level) from the `sprt` table and aliasing it as "livello". We use an `INNER JOIN` to join the `sprt` table with the `prtct` table on the `nm` and `lvll` columns, which represent the sport name and level respectively. The `WHERE` clause is used to filter only sports that are practiced indoors (`spz = 'indoor'`).

*Il modello coglie correttamente il fatto che con "praticati al chiuso" ci si riferisce a "indoor" valore assunto dall'attributo spz(spazio). Si poteva ottenere tutto dalla tabella prtct.

CHAT GPT

Query 3 - ugly & insert

Livello : 1.5

Esito: ***Corretto**

f

Traduci in SQL la seguente query "Seleziona nome e livello degli sport praticati al chiuso"



Per selezionare il nome e il livello degli sport praticati al chiuso dalla tabella `'prtct'`, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT sprt, lvll  
FROM prtct  
WHERE spz = 'indoor';
```

Questa query selezionerà i campi `'sprt'` e `'lvll'` dalla tabella `'prtct'` dove il valore del campo `'spz'` è "indoor", che corrisponde agli sport praticati al chiuso.

*Il modello coglie correttamente il fatto che con "praticati al chiuso" ci si riferisce a "indoor" valore assunto dall'attributo spz(spazio). La query ottiene tutto dalla tabella prtct.

CHAT GPT

Query 4

Livello : 1.5

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"

Traduci la seguente query in SQL:

sql

Copy code

```
SELECT nome, cognome, telefono, email, professione  
FROM specialista  
ORDER BY professione;
```

Questa query seleziona il nome, il cognome, i contatti (telefono e email) e la professione di ogni specialista (sanitario) dalla tabella `specialista` e li ordina in base alla colonna `professione`. Gli specialisti saranno elencati in ordine crescente secondo il valore della colonna `professione`.

*Chiedendo il nome e cognome di ogni sanitario ci si riferisce a medici o curanti(nel nostro db Specialisti). Quindi non andava coinvolta la tabella CentroSanitario. La parola “sanitario” che in questo caso è sinonimo di specialista introduce un’ambiguità che viene ben gestita dal modello.

CHAT GPT

Query 4- schema ugly

Livello : 1.5

Esito: Corretto



Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"



Based on the provided database model, the SQL query to select the name, surname, contacts, and profession of each healthcare professional ordered by their profession would be as follows:



```
sql Copy code
SELECT nm AS nome, cgnm AS cognome, tlfn AS contatti, qlfc AS professione
FROM prsnlmmnstrtv
ORDER BY qlfc;
```

In this query, we are selecting the fields `nm` (name) from the `prsnlmmnstrtv` table and aliasing it as "nome", the field `cgnm` (surname) and aliasing it as "cognome", the field `tlfn` (contacts) and aliasing it as "contatti", and the field `qlfc` (profession) and aliasing it as "professione". The table `prsnlmmnstrtv` seems to contain information about healthcare professionals. We use the `ORDER BY` clause to sort the results based on the `qlfc` column, which represents the profession of each healthcare professional. The result will be a list of healthcare professionals sorted by their profession.

*Chiedendo il nome e cognome di ogni sanitario ci si riferisce a medici o curanti(nel nostro db Spclst cioè Specialisti).

Quindi non andava coinvolta la tabella

Prsnlmmnstrtv(PersonaleAmministrativo).

Non solo la parola ambigua "sanitario" porta il modello a sbagliare ma anche

"professione" porta il modello a considerare la "qualifica"(e quindi il campo qlfc) del personale amministrativo.

CHAT GPT

Query 4 - ugly & insert

Livello : 1.5

Esito: ***Scorretto**



Traduci in SQL la seguente query: "Seleziona nome, cognome, contatti e professione di ogni sanitario ordinati per professione"



Per selezionare il nome, cognome, contatti e professione di ogni sanitario ordinati per professione dalla tabella `scrtt`, `prsnlmmnstrtv` e `prsnltcnc`, puoi utilizzare la seguente query SQL con una clausola JOIN e un'istruzione ORDER BY:

sql

Copy code

```
SELECT s.nm, s.cgnm, s.ndrzz, s.ctt, s.tlfn, s.ml, p.prfssn AS professione
FROM scrtt s
LEFT JOIN prsnlmmnstrtv p ON s.mtrcl = p.mtrcl
LEFT JOIN prsnltcnc c ON s.mtrcl = c.mtrcl
ORDER BY professione;
```



Questa query unirà le tabelle `scrtt`, `prsnlmmnstrtv` e `prsnltcnc` usando le clausole LEFT JOIN e selezionerà i campi `nm`, `cgnm`, `ndrzz`, `ctt`, `tlfn`, `ml` dalla tabella `scrtt` e il campo `prfssn` dalla tabella `prsnlmmnstrtv` (che rappresenta la professione) aliasata come "professione". Infine, i risultati saranno ordinati per professione.

*Chiedendo il nome e cognome di ogni sanitario ci si riferisce a medici o curanti(nel nostro db Spclst cioè Specialisti). Quindi non andava coinvolta la tabella Prsnlmmnstrtv(Personale Amministrativo).

CHAT GPT

Query 5

Livello : 2

Esito: ***Scorretto**



Per ottenere la matricola, nome, cognome e ruolo dei membri del personale amministrativo con diploma che hanno un impiego nel centro sportivo "Appia Sport Village", è necessario utilizzare una query SQL con una JOIN tra le tabelle '`personaleamministrativo`', '`impiego`' e '`centrosportivo`'. La condizione di filtro sarà basata sulla corrispondenza dell'ID del centro sportivo nel quale lavora il membro del personale amministrativo con il nome del centro specificato.



Ecco la query:

```
sql Copy code
SELECT pa.matricola, pa.nome, pa.cognome, pa.qualifica AS ruolo
FROM personaleamministrativo AS pa
JOIN impiego AS i ON pa.matricola = i.matricola
JOIN centrosportivo AS cs ON i.idCentro = cs.id
WHERE pa.titolo = 'diploma' AND cs.nome = 'Appia Sport Village';
```

Questa query otterrà i dati richiesti per i membri del personale amministrativo con diploma impiegati nel centro sportivo "Appia Sport Village". Assicurati di utilizzare il nome corretto del centro sportivo nelle condizioni WHERE.

***idCentro** è un campo di `PersonaleAmministrativo`. È sbagliato collegare `PersonaleAmministrativo` con `impiego.Impiego` è una relazione tra centri sportivi e `PersonaleTecnico`

CHAT GPT

Query 5 - schema ugly

Livello : 2

Esito: *Scorretto

Based on the provided database model, to translate the given query "matricola, nome, cognome, ruolo" (matriculation number, name, surname, role) of administrative staff members with a diploma who are employed at the sports center "Appia Sport Village," you would use the following SQL query:

```
sql
Copy code

SELECT prsnlmmnstrtv.mtrcl AS matricola, prsnlmmnstrtv.nm AS nome, prsnlmmnstrtv.cgnm AS cognome, prsnlmmnstrtv.qlfc AS ruolo
FROM prsnlmmnstrtv
INNER JOIN scrtt ON prsnlmmnstrtv.mtrcl = scrtt.cdcCrtfct
WHERE prsnlmmnstrtv.ttl = 'diploma' AND scrtt.ndrzz = 'Appia Sport Village';
```

In this query, we are selecting the fields `mtrcl` (matriculation number) from the `prsnlmmnstrtv` table and aliasing it as "matricola", the field `nm` (name) and aliasing it as "nome", the field `cgnm` (surname) and aliasing it as "cognome", and the field `qlfc` (role) and aliasing it as "ruolo" from the table `prsnlmmnstrtv`. The table `prsnlmmnstrtv` seems to contain information about administrative staff members.

We then perform an `INNER JOIN` with the `scrtt` table on the `mtrcl` (matriculation number) column from both tables. This is based on the assumption that the field `cdcCrtfct` in the `scrtt` table represents the matriculation number of the administrative staff member.

The `WHERE` clause is used to filter the results, specifying that we want only staff members with a diploma (`ttl = 'diploma'`) and who are employed at the sports center "Appia Sport Village" (`scrtt.ndrzz = 'Appia Sport Village'`).

*Completamente Sbagliata.
Tanti Errori.
Si considera erroneamente la tabella scrtt(Iscritto).
Viene effettuato un join tra prsnlmmnstrtv(PersonaleAmministrativo) e scrtt(Iscritto) confrontando certificati medici e matricole.

CHAT GPT

Query 5 - ugly & insert

Livello : 2

Esito: Corretto

f Traduci in sql la seguente query :"matricola, nome, cognome, ruolo del membro del personale amministrativo con diploma che ha un impiego nel centro sportivo "Appia Sport Village"



Per tradurre la richiesta in SQL, dovremo utilizzare una query SELECT per selezionare i dati richiesti dalla tabella '`prsnlmmnstrtv`' (membri del personale amministrativo con diploma) e la tabella '`cntrsprtv`' (centro sportivo "Appia Sport Village"). Dovremo inoltre unire queste due tavelle in base alla corrispondenza tra il campo '`dCntr`' nella tabella '`prsnlmmnstrtv`' e il campo '`id`' nella tabella '`cntrsprtv`', in modo da ottenere i dati dei membri del personale amministrativo impiegati nel centro sportivo specificato.



La query SQL sarà la seguente:

```
sql
SELECT prsnlmmnstrtv.mtrcl AS matricola, prsnlmmnstrtv.nm AS nome, prsnlmmnstrtv.cgnm AS cognome, prsnlmmnstrtv.qlfc AS ruolo
FROM prsnlmmnstrtv
INNER JOIN cntrsprtv ON prsnlmmnstrtv.dCntr = cntrsprtv.id
WHERE prsnlmmnstrtv.ttl = 'diploma' AND cntrsprtv.nm = 'Appia Sport Village'
```

Questo query selezionerà i campi '`mtrcl`', '`nm`', '`cgnm`', e '`qlfc`' dalla tabella '`prsnlmmnstrtv`' (con gli alias '`matricola`', '`nome`', '`cognome`', e '`ruolo`', rispettivamente) per i membri del personale amministrativo con il titolo '`diploma`' che sono impiegati nel centro sportivo "Appia Sport Village" dalla tabella '`cntrsprtv`'.

CHAT GPT

Query 6

Livello : 2

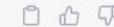
Esito: Corretto

f

Traduci in sql la seguente query : "nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019"



Per ottenere il nome, cognome e telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019, è necessario utilizzare una JOIN tra le tabelle 'iscritto' e 'donazione' basata sul campo 'codiceFiscale', in modo da collegare gli iscritti alle donazioni effettuate.



Ecco la query SQL:

```
sql Copy code
SELECT i.nome, i.cognome, i.telefono
FROM iscritto AS i
JOIN donazione AS d ON i.codiceFiscale = d.codiceFiscale
WHERE d.data > '2019-05-01';
```

Questa query otterrà i dati richiesti per gli iscritti che hanno effettuato almeno una donazione dopo la data specificata. Assicurati di utilizzare il formato corretto per la data nella clausola WHERE (AAAA-MM-GG).

CHAT GPT

Query 6 -schema ugly

Livello : 2

Esito: ***Scorretto**



Based on the provided database model, to translate the given query "nome, cognome, telefono" (name, surname, phone) of members who have made at least one donation after the date 01/05/2019, you would use the following SQL query:

sql

Copy code

```
SELECT tnt.nm AS nome, tnt.cgnm AS cognome, tnt.t AS telefono
FROM tnt
INNER JOIN dnzn ON tnt.id = dnzn.dCntr
WHERE dnzn.dt > '2019-05-01';
```

In this query, we are selecting the fields `'nm'` (name) from the `'tnt'` table and aliasing it as "nome", the field `'cgnm'` (surname) and aliasing it as "cognome", and the field `'t'` (phone) and aliasing it as "telefono" from the table `'tnt'`. The table `'tnt'` seems to contain information about the members.

We then perform an `'INNER JOIN'` with the `'dnzn'` table on the `'id'` (member ID) column from the `'tnt'` table and the `'dCntr'` (member ID) column from the `'dnzn'` table. This is based on the assumption that the field `'id'` in the `'tnt'` table represents the member ID, and the field `'dCntr'` in the `'dnzn'` table represents the member ID for whom the donation is made.

The `'WHERE'` clause is used to filter the results, specifying that we want only members who have made at least one donation after the date 01/05/2019 (`'dnzn.dt > '2019-05-01'`).

The date format used is assumed to be 'YYYY-MM-DD'.

***Completamente sbagliata.**
Si coinvolge erroneamente la tabella `tnt(Utente)`.
Inoltre il join si effettua comparando l'id del centro donazione e l'id dell'utente.
Invece il join andava fatt su codice fiscale del donatore e codice fiscale dell'iscritto.

CHAT GPT

Query 6 - ugly & insert

Livello : 2

Esito: ***Scorretto**

 Traduci in sql la seguente query : "nome,cognome,telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019"

 Per ottenere i dati richiesti, possiamo utilizzare una query SQL che seleziona il nome, cognome e telefono degli iscritti che hanno effettuato almeno una donazione dopo la data 01/05/2019. La query sarà la seguente:

```
sql  Copy code  
  
SELECT s.nm AS nome, s.cgnm AS cognome, s.tlfn AS telefono  
FROM scrtt s  
INNER JOIN dnzn d ON s.cdcFsc1 = d.cdcFsc1  
WHERE d.dt > '2019-05-01';
```

In questa query, stiamo selezionando i campi "nm" e "cgnm" dalla tabella "scrtt" (che rappresenta gli iscritti) e il campo "tlfn" dalla tabella "dnzn" (che rappresenta le donazioni). Utilizziamo l'operatore JOIN per combinare le due tabelle in base al campo "cdcFsc1", che è comune tra le tabelle.

La clausola WHERE specifica che vogliamo solo le righe in cui la data della donazione (d.dt) sia successiva al 1 maggio 2019. In questo modo, otterremo solo gli iscritti che hanno effettuato almeno una donazione dopo quella data.

CHAT GPT

Query 7

Livello : 2

Esito: *Scorretto



Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"



Per calcolare la media delle recensioni di ogni centro sportivo, dovrà utilizzare una query SQL con la funzione di aggregazione `'AVG()'` per calcolare la media e il raggruppamento `'GROUP BY'` per ottenere il risultato per ogni centro sportivo.



Supponendo che il punteggio delle recensioni sia memorizzato nel campo `'struttura'` nella tabella `'recensione'`, e il campo `'idCentro'` rappresenti l'ID del centro sportivo, la query sarà la seguente:

sql

Copy code

```
SELECT idCentro, AVG(struttura) AS media_recensioni
FROM recensione
GROUP BY idCentro;
```

Questo query calcolerà la media delle recensioni di ogni centro sportivo e restituirà due colonne: `'idCentro'`, che rappresenta l'ID del centro sportivo, e `'media_recensioni'`, che conterrà il valore medio delle recensioni per ciascun centro.

*Nel calcolo delle medie recensioni andavano coinvolte anche i campi staffTecnico e gestione oltre a struttura.

CHAT GPT

Query 7 -schema ugly

Livello : 2

Esito: *Corretto

Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"

Based on the provided database model, to calculate the average of reviews for each sports center, you would use the following SQL query:

```
sql
SELECT dCntr, AVG(strttr + gstn + stffTcnc) AS media_recensioni
FROM rcnsn
GROUP BY dCntr;
```

In this query, we are selecting the `dCntr` (center ID) column from the `rcnsn` table and using the `AVG` function to calculate the average of the sum of the columns `strttr` (review rating for structure), `gstn` (review rating for guest treatment), and `stffTcnc` (review rating for staff and coaching). We are aliasing the calculated average as "media_recensioni".

We use the `GROUP BY` clause to group the results based on the `dCntr` column, which represents the center ID. This allows us to calculate the average of reviews for each sports center separately.

The result of this query will give you the average review rating for each sports center based on the individual review ratings for structure, guest treatment, and staff and coaching.

*Nella query si calcola la media della somma dei voti relativi a struttura,staffTecnico e gestione per ogni centro sportivo.

Sono coinvolti i campi giusti.

Anche se non si chiede esattamente (il modello sta attribuendo lo stesso peso ad ogni voto senza che gli viene chiesto) questo tipo di calcolo il risultato è comunque apprezzabile e quindi corretto.

CHAT GPT

Query 7 - ugly & schema

Livello : 2

Esito: *Scorretto



Traduci in SQL la seguente query: "Calcola la media delle recensioni di ogni centro sportivo"



Per calcolare la media delle recensioni di ogni centro sportivo, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT
    dCntrSprtv AS idCentroSportivo,
    AVG(gstn) AS mediaRecensioni
FROM rcnsn
GROUP BY dCntrSprtv;
```

Questa query selezionerà il `dCntrSprtv` (l'ID del centro sportivo) dalla tabella `rcnsn`, quindi calcolerà la media delle recensioni per ogni centro sportivo utilizzando la funzione AVG.

Regenerate response

*La struttura della query è corretta ma vengono ignorati campi importanti come struttura e staffTecnico.

CHAT GPT

Query 8

Livello : 2

Esito: *Corretto

f

Traduci in SQL la seguente query: "Seleziona il nome,cognome,centro,indirizzo e contatti degli osteopati e degli ortopedici che lavorano a Roma"



Per ottenere il nome, cognome, centro, indirizzo e contatti degli osteopati e degli ortopedici che lavorano a Roma, è necessario utilizzare una query SQL con una JOIN tra le tabelle `specialista`, `centrosanitario` e `personaletecnico`, basata sulle professioni degli specialisti e sulla città in cui si trova il centro sanitario.



Supponendo che il campo `professione` nella tabella `specialista` contenga le informazioni sulla professione (come "osteopata" e "ortopedico"), il campo `idCentro` rappresenti l'ID del centro sanitario nella tabella `specialista`, e il campo `citta` nella tabella `centrosanitario` rappresenti la città in cui si trova il centro, la query sarà la seguente:

sql

Copy code

```
SELECT s.nome, s.cognome, cs.nome AS centro, cs.indirizzo, s.telefono, s.email
FROM specialista AS s
JOIN centrosanitario AS cs ON s.idCentro = cs.id
WHERE s.professione IN ('osteopata', 'ortopedico') AND cs.citta = 'Roma';
```

Questo query restituirà i dati richiesti per gli osteopati e gli ortopedici che lavorano a Roma, inclusi il nome, cognome, nome del centro, indirizzo e contatti (telefono ed email) degli

*anche se non abbiamo esplicitato, il tipo di Centro che ci interessa nella query, il modello individua Centro Sanitario correttamente

CHAT GPT

Query 8 -schema ugly

Livello : 2

Esito: *Scorretto

La seguente query SQL seleziona il nome, cognome, centro, indirizzo e contatti degli osteopati e degli ortopedici che lavorano a Roma:

```
sql
SELECT
    prsnlmmnstrtv.nm AS nome,
    prsnlmmnstrtv.cgnm AS cognome,
    cntrsntc.nm AS centro,
    cntrsntc.ndrzz AS indirizzo,
    cntrsntc.tlfn AS telefono,
    cntrsntc.ml AS email
FROM
    prsnlmmnstrtv
INNER JOIN
    prsnltcnc ON prsnlmmnstrtv.mtrcl = prsnltcnc.mtrcl
INNER JOIN
    cntrsntc ON prsnltcnc.dCntr = cntrsntc.id
WHERE
    (prsnltcnc.qlfc = 'osteopata' OR prsnltcnc.qlfc = 'ortopedico')
    AND cntrsntc.ctt = 'Roma';
```

In questa query, utilizziamo tre tabelle:

1. `'prsnlmmnstrtv'`: Tabella che contiene informazioni sugli osteopati e ortopedici, con i campi `'nm'` (nome) e `'cgnm'` (cognome).
2. `'prsnltcnc'`: Tabella che associa i medici nella tabella `'prsnlmmnstrtv'` ai centri nella tabella `'cntrsntc'`. Contiene il campo `'qlfc'`, che identifica la qualifica dell'osteopata o ortopedico.
3. `'cntrsntc'`: Tabella che contiene informazioni sui centri, inclusi il nome (`'nm'`), l'indirizzo (`'ndrzz'`), il telefono (`'tlfn'`) e l'email (`'ml'`).

Utilizzando gli `'INNER JOIN'` collegiamo le tabelle appropriate, quindi utilizziamo la clausola `'WHERE'` per filtrare i risultati in base alla qualifica e alla città del centro.

*Tanti errori.
Oltre a considerare ortopedico e osteopata come qualifiche del personale tecnico considera quest'ultimo come impiegato dei Centri Sanitari.

CHAT GPT

Query 8 - ugly & insert

Livello : 2

Esito: Corretto

f Traduci in SQL la seguente query: "Seleziona il nome,cognome,centro,indirizzo e contatti degli osteopati e degli ortopedici che lavorano a Roma "

Per selezionare il nome, cognome, centro, indirizzo e contatti degli osteopati e degli ortopedici che lavorano a Roma, puoi utilizzare la seguente query SQL:

```
sql
Copy code
SELECT
    s.nm AS nome,
    s.cgnm AS cognome,
    c.nm AS centro,
    c.ndrzz AS indirizzo,
    c.tlfn AS telefono,
    c.mil AS email
FROM
    spclst s
JOIN
    cntrsntz c ON s.dCntr = c.id
WHERE
    s.prfssn IN ('osteopata', 'ortopedico') AND c.ctt = 'Roma';
```

Questa query utilizza una combinazione di JOIN tra le tabelle `'spclst'` e `'cntrsntz'` sulla colonna `'dCntr'`, che collega i professionisti sanitari ai rispettivi centri sportivi in cui lavorano. La clausola `'WHERE'` filtra solo gli osteopati e gli ortopedici che lavorano a Roma, in base alla colonna `'ctt'` della tabella `'cntrsntz'`. Il risultato mostrerà il nome, cognome, centro, indirizzo, telefono ed email degli osteopati e ortopedici che soddisfano i criteri specificati.

*Corretto.
Ortopedico e
Osteopata sono
effettivamente
specialisti e lavorano
in centri sanitari.
NOTA: nelle insert
non sono presenti
“ortopedico” e
“osteopata” ma sono
presenti cose simili
tipo “fisioterapista”...

CHAT GPT

Query 9

Livello : 3

Esito:Corretto

f

Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"



Per selezionare la data della visita, il nome, il cognome e il codice fiscale del visitato, insieme alla professione del visitante, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT v.data AS "data visita", i.nome, i.cognome, i.codiceFiscale, s.professione
FROM visita v
JOIN iscritto i ON v.codiceFiscale = i.codiceFiscale
JOIN specialista s ON v.matricolaSpecialista = s.matricola;
```

In questa query, stiamo selezionando i campi `data` dalla tabella `visita` rinominandoli come "data visita". Utilizziamo la clausola `JOIN` per collegare la tabella `visita` con le tabelle `iscritto` e `specialista`, in modo da ottenere le informazioni sul visitato (nome, cognome e codice fiscale) dalla tabella `iscritto` e la professione del visitante dalla tabella `specialista` utilizzando le corrispondenti chiavi esterne (`codiceFiscale` e `matricolaSpecialista`). In questo modo, otterrai i dettagli delle visite insieme alle informazioni sul visitato e la professione del visitante.

CHAT GPT

Query 9 -schema ugly

Livello : 3

Esito:***Scorretto**

Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"

Per tradurre la query in SQL, è necessario avere informazioni sulle tabelle coinvolte nel database. Supponendo che ci siano tabelle denominate '**visite**' per memorizzare le informazioni sulle visite e '**visitatori**' per memorizzare le informazioni sui visitatori, ecco come sarebbe la query SQL:

```
sql
SELECT
    v.dt AS "data visita",
    vst.nm AS nome,
    vst.cgnm AS cognome,
    vst.cdcFscl AS codiceFiscale,
    prsnltcnc qlfc AS professione
FROM
    visite v
JOIN
    visitatori vst ON v.idVisitato = vst.id
JOIN
    prsnlmmnstrtv prsnltcnc ON v.idVisitante = prsnltcnc.mtrcl;
```

In questa query:

- Supponiamo che nella tabella delle visite ci sia un campo chiamato '**dt**' che rappresenta la data della visita.
- Nella tabella '**visitatori**' ci sono i campi '**nm**' (nome), '**cgnm**' (cognome) e '**cdcFscl**' (codice fiscale) del visitato.
- Nella tabella '**prsnlmmnstrtv**' ci sono informazioni sui visitanti, inclusa la loro professione identificata dal campo '**qlfc**'.
- Usiamo gli '**INNER JOIN**' per collegare le tabelle in base alle corrispondenti chiavi primarie

***Anziché considerare la tabella Spclst(Specialista) per Visitante si considerano erroneamente le tabelle Prsnlmmnstrtv(PersonaleAmm) e Prsnltcntc(PersonaleTec) in maniera tra l'altro molto confusa.**

CHAT GPT

Query 9 - ugly & insert

Livello : 3

Esito:***Scorretto**

f Traduci in SQL la seguente query: "data visita,(nome,cognome e codiceFiscale del visitato) e professione del visitante"



Per ottenere i dati richiesti nella query in SQL, possiamo utilizzare la clausola JOIN per collegare le tabelle coinvolte. Di seguito, viene fornita la query SQL:



```
sql Copy code
SELECT vst.dt AS 'data visita',
       scrtt.nm AS 'nome',
       scrtt.cgnm AS 'cognome',
       scrtt.cdcFscl AS 'codiceFiscale del visitato',
       prsnlmmnstrtv.prfssn AS 'professione del visitante'
FROM vst
JOIN spclst ON vst.mtrclSpclst = spclst.mtrcl
JOIN scrtt ON vst.cdcFscl = scrtt.cdcFscl
JOIN prsnlmmnstrtv ON spclst.mtrcl = prsnlmmnstrtv.mtrcl;
```

Questa query selezionerà la data della visita dalla tabella '**vst**', il nome, cognome e codice fiscale del visitato dalla tabella '**scrtt**', e la professione del visitante dalla tabella '**prsnlmmnstrtv**', collegando le tabelle attraverso le chiavi esterne appropriate. Assicurati di avere correttamente impostato le relazioni tra le tabelle nel tuo database.

*Anziché considerare solo la tabella Spclst(Specialista) per Visitante si considera erroneamente anche la tabella Prsnlmmnstrtv(PersonaleAmm).

CHAT GPT

Query 10

Livello : 3.5

Esito: Corretto



Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."



Per ottenere il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e il nome di questi sport, è necessario utilizzare una JOIN tra le tabelle `centrosportivo` e `praticato` basata sull'ID del centro sportivo e il campo `idCentro`, rispettivamente. Inoltre, sarà necessario aggiungere una condizione per selezionare solo gli sport a livello dilettantistico. Infine, utilizzeremo l'istruzione ORDER BY per ordinare i record in ordine alfabetico degli sport.



Assumendo che il campo `livello` nella tabella `praticato` contenga le informazioni sul livello dello sport (come "dilettantistico" e "professionistico"), e il campo `idCentro` nella tabella `praticato` rappresenti l'ID del centro sportivo, la query sarà la seguente:

sql

Copy code

```
SELECT cs.nome AS nome_centro, cs.telefono, cs.email, p.sport
FROM centrosportivo AS cs
JOIN praticato AS p ON cs.id = p.idCentro
WHERE p.livello = 'dilettantistico'
ORDER BY p.sport;
```

Questo query restituirà i dati richiesti per i Centri Sportivi in cui si praticano sport a livello dilettantistico, includendo il nome del centro, il telefono, l'email e il nome degli sport dilettantistici praticati. I risultati saranno ordinati alfabetico degli sport. Assicurati che i nomi dei campi e delle tabelle siano corretti e presenti nel tuo database.

*Il modello seleziona correttamente i campi email e telefono in quanto "contatti"

CHAT GPT

Query 10 -schema ugly

Livello : 3.5

Esito: *Scorretto

f Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."

Per tradurre la query in SQL, assumiamo l'esistenza delle seguenti tabelle:

1. `cntrsprt` : Tabella che contiene informazioni sui centri sportivi, inclusi il nome (`nm`) e i contatti (ad esempio, telefono, email, ecc.).
2. `cmptnz` : Tabella che contiene informazioni sugli sport praticati nei centri sportivi, incluso il nome dello sport (`sppt`) e il livello (`lvll`) a cui viene praticato (dilettantistico o professionistico).

La query SQL sarebbe la seguente:

```
sql Copy code
SELECT
    cntrsprt.nm AS nome_centro,
    cntrsprt.tlfn AS telefono,
    cntrsprt.mil AS email,
    cmptnz.sppt AS sport_praticato
FROM
    cntrsprt
JOIN
    cmptnz ON cntrsprt.id = cmptnz.dCntr
WHERE
    cmptnz.lvll = 'dilettantistico'
ORDER BY
    cmptnz.sppt ASC;
```

In questa query:

1. Utilizziamo gli `INNER JOIN` per collegare le tabelle `cntrsprt` e `cmptnz` in base al campo `id` del centro sportivo e al campo `dCntr` nella tabella `cmptnz`, che rappresenta la chiave esterna che lega gli sport ai centri sportivi.
2. Nella clausola `WHERE`, filtriamo solo gli sport a livello dilettantistico.

*sarebbe più sensato passare per la tabella Prtct(Praticato) che riguarda Sport e Centri piuttosto che Cmptz(Competenza) che riguarda sport e personale tecnico. Inoltre non c'è un personaleTecnico per ogni sport .Quindi i risultati della query non saranno completi.

CHAT GPT

Query 10 - ugly & insert

Livello : 3.5

Esito: ***Scorretto**

f Traduci in SQL la seguente query: "Seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e nome di questi sport. La tabella generata dovrà mostrare i record ordinati in ordine alfabetico degli sport."

Ecco la query SQL che seleziona il nome e i contatti dei Centri Sportivi in cui si praticano sport a livello dilettantistico e il nome di questi sport, ordinati in ordine alfabetico degli sport:

```
sql
Copy code

SELECT
    cntrsntr.nm AS "Nome Centro Sportivo",
    cntrsntr.ndrzz AS "Indirizzo",
    cntrsntr.ctt AS "Città",
    cntrsntr.tlfn AS "Telefono",
    cntrsntr.ml AS "Email",
    sprt.nm AS "Sport Dilettantistico"
FROM
    cntrsntr
JOIN
    prtct ON cntrsntr.id = prtct.dCntr
JOIN
    sprt ON prtct.spkt = sprt.nm AND prtct.lvll = 'dilettantistico'
ORDER BY
    sprt.nm;
```

Questa query utilizza le tabelle 'cntrsntr', 'prtct', e 'sprt' per ottenere il nome, gli indirizzi, la città, il telefono e l'email dei centri sportivi in cui si praticano sport a livello dilettantistico. Inoltre, elenca gli sport dilettantistici ordinati in ordine alfabetico degli sport.

Regenerate response

*Viene confuso
CentroSanitario(cntrsnt
r) con
CentroSportivo(cntrsprt
v).

CHAT GPT

Query 11

Livello : 3.5

Esito: *Scorretto

f matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale



Per ottenere la matricola, sport, livello e centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi non situati nella capitale, dovrai utilizzare una query SQL con una JOIN tra le tabelle 'personaletecnico', 'praticato' e 'centrosportivo'. Inoltre, useremo la condizione 'WHERE' per escludere i centri sportivi della capitale dalla selezione.

Assumendo che il campo 'matricola' nella tabella 'personaletecnico' rappresenti l'ID del membro del personale tecnico, il campo 'sport' nella tabella 'praticato' rappresenti lo sport praticato, e il campo 'livello' nella tabella 'praticato' rappresenti il livello dello sport (come "dilettantistico" e "professionistico"), e il campo 'idCentro' nella tabella 'personaletecnico' rappresenti l'ID del centro sportivo in cui lavora il membro del personale, e il campo 'citta' nella tabella 'centrosportivo' rappresenti la città in cui si trova il centro sportivo, la query sarà la seguente:

```
sql
Copy code

SELECT pt.matricola, pr.sport, pr.livello, cs.nome AS centro_sportivo
FROM personaletecnico AS pt
JOIN praticato AS pr ON pt.matricola = pr.matricola
JOIN centrosportivo AS cs ON pr.idCentro = cs.id
WHERE cs.citta <> 'Capitale';
```

Questo query restituirà i dati richiesti per i membri del personale tecnico che lavorano nei centri sportivi situati in città diversi. Regenerate response

*Il modello deve passare attraverso la tabella Impiego che mette in relazione PersonaleTecnico e CentroSportivo. Inoltre con "NON della capitale" si fa riferimento a centri sportivi la cui città è diversa da Roma.

CHAT GPT

Query 11 -schema ugly

Livello : 3.5

Esito: *Scorretto

Traduci in SQL la seguente query: "matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale"

Per tradurre la query in SQL, assumiamo l'esistenza delle seguenti tabelle:

1. **'prsnltcnc'**: Tabella che contiene informazioni sul personale tecnico, incluso il campo **'mtrcl'** (matricola) e **'qlfc'** (livello, ad esempio, istruttore o preparatore).
2. **'prtct'**: Tabella che contiene informazioni sugli sport praticati dai membri del personale tecnico, incluso il campo **'sprt'** (sport) e **'lvll'** (livello, ad esempio, dilettantistico o professionistico).
3. **'cntrsprtvt'**: Tabella che contiene informazioni sui centri sportivi, incluso il campo **'id'** (identificativo del centro) e la città in cui si trova il centro.

La query SQL sarebbe la seguente:

```
sql
SELECT
    prsnltcnc.mtrcl AS matricola,
    prtct.sppt AS sport,
    prtct.lvll AS livello,
    cntrsprtvt.nm AS centro_sportivo
FROM
    prsnltcnc
JOIN
    prtct ON prsnltcnc.mtrcl = prtct.mtrcl
JOIN
    cntrsprtvt ON prtct.dCntr = cntrsprtvt.id
WHERE
    cntrsprtvt.ctt <> 'Roma';
```

In questa query:

*Collega erroneamente la tabella **prtct(praticato)** e **prsnltcnc(personale tecnico)**. Coglie il significato di “non della capitale”.

CHAT GPT

Query 11 - ugly & insert

Livello : 3.5

Esito: ***Scorretto**

f Traduci in SQL la seguente query: "matricola,sport,livello,centro sportivo dei membri del personale tecnico che lavorano nei centri sportivi NON della capitale"

*Confonde personale amministrativo(prsnlmmnstrtv) con personale tecnico(prsntcnc).

Per ottenere i membri del personale tecnico che lavorano nei centri sportivi NON della capitale, possiamo utilizzare la seguente query SQL:

```
sql
SELECT prsnlmmnstrtv.mtrcl AS matricola,
       prtct.sprt AS sport,
       prtct.lvll AS livello,
       cntrsnnr.nm AS centro_sportivo
  FROM prsnlmmnstrtv
 JOIN prtct ON prsnlmmnstrtv.mtrcl = prtct.mtrcl
 JOIN cntrsnnr ON prsnlmmnstrtv.dCntr = cntrsnnr.id
 WHERE cntrsnnr.ctt <> 'Roma';
```

CHAT GPT

Query 12

Livello : 3

Esito:***Scorretto**

f

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione"



Per selezionare il nome e l'ID dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data in cui è iniziata la convenzione, possiamo utilizzare una query SQL con una combinazione di JOIN tra le tabelle `centrosportivo` e `centrodonazione` con le tabelle di collegamento `convenzione_cs` e `convenzione_cd`.



Ecco come appare la query:

```
sql
SELECT
    cs.id AS centro_sportivo_id,
    cs.nome AS centro_sportivo_nome,
    cd.id AS centro_donazione_id,
    cd.nome AS centro_donazione_nome,
    cc.data AS data_inizio_convenzione_cs,
    cd.data AS data_inizio_convenzione_cd
FROM
    centrosportivo cs
LEFT JOIN
    convenzione_cs cc ON cs.id = cc.idCentroSportivo
LEFT JOIN
    centrodonazione cd ON cc.idCentroDonazione = cd.id
LEFT JOIN
    convenzione_cd cd ON cs.id = cd.idCentroSportivo;
```

***Viene coinvolta erroneamente la tabella convenzione_cs che non riguarda i Centri Sportivi o Centri Donazione ma i Centri Sanitari**

CHAT GPT

Query 12 -schema ugly

Livello : 3

Esito:***Scorretto**

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione".

Per tradurre la query in SQL, assumiamo l'esistenza delle seguenti tabelle:

Supponiamo anche che entrambe le tabelle di relazione ('cnvnzn_cd' e 'cnvnzn_cs') abbiano un campo 'dt' che rappresenta la data in cui inizia la convenzione tra i centri sportivi e i centri di donazione.

La query SQL sarebbe la seguente:

```
SELECT
    cntrsprtv.id AS id_centro_sportivo,
    cntrsprtv.nm AS nome_centro_sportivo,
    cntrdnzn.id AS id_centro_donazione,
    cntrdnzn.nm AS nome_centro_donazione,
    cnvnzn_cd.dt AS data_inizio_convenzione
FROM
    cntrsprtv
JOIN
    cnvnzn_cd ON cntrsprtv.id = cnvnzn_cd.dCntrSprtv
JOIN
    cntrdnzn ON cnvnzn_cd.dCntrDnzn = cntrdnzn.id
UNION
SELECT
    cntrsprtv.id AS id_centro_sportivo,
    cntrsprtv.nm AS nome_centro_sportivo,
    cntrdnzn.id AS id_centro_donazione,
    cntrdnzn.nm AS nome_centro_donazione,
    cnvnzn_cs.dt AS data_inizio_convenzione
FROM
    cntrsprtv
JOIN
    cnvnzn_cs ON cntrsprtv.id = cnvnzn_cs.dCntrSprtv
JOIN
    cntrdnzn ON cnvnzn_cs.dCntrSntz = cntrdnzn.id;
```

In questa query:

***La prima Select è corretta mentre la seconda che coinvolge le convenzioni con i centri sanitari porta a risultati errati**

CHAT GPT

Query 12 - ugly & insert

Livello : 3

Esito:Corretto

Traduci in SQL la seguente query: "Seleziona nome ed id dei centri sportivi e dei centri donazione con i quali sono convenzionati e la data a cui risale l'inizio della convenzione"

La seguente query SQL selezionerà il nome, l'id e la data di inizio della convenzione dei centri sportivi e dei centri donazione con i quali sono convenzionati:

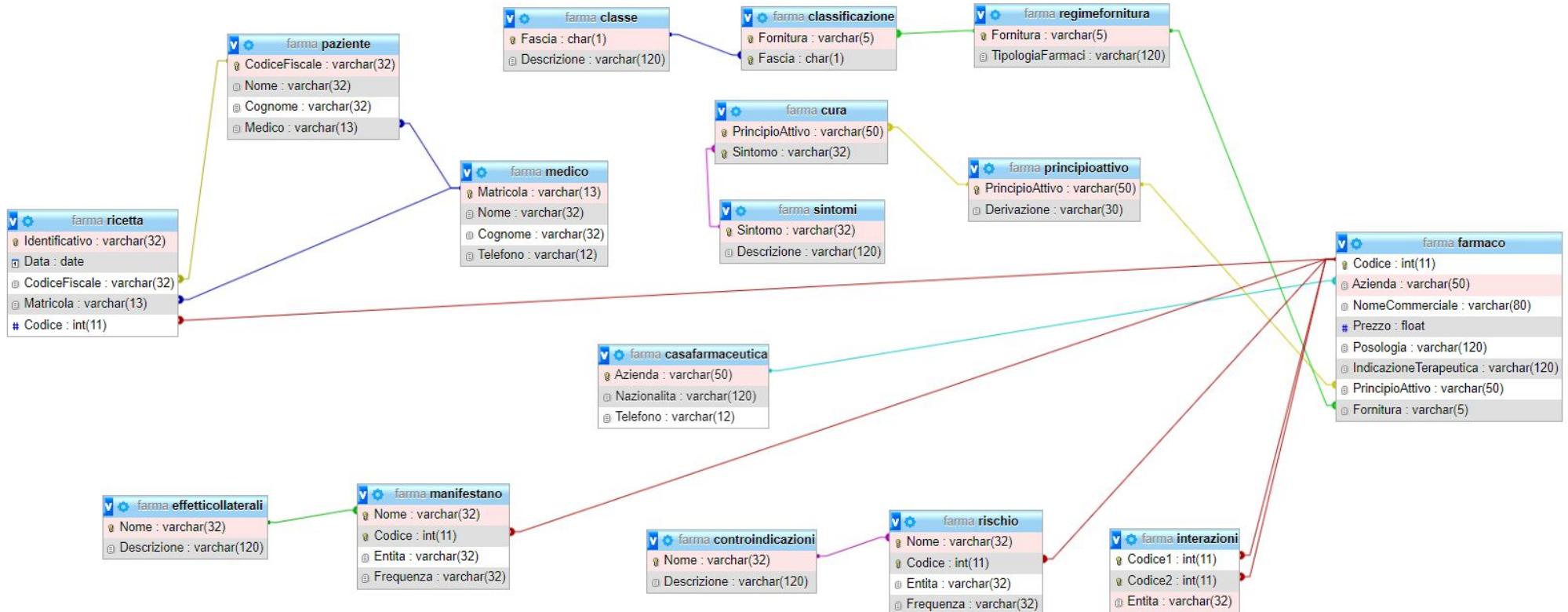
```
sql
Copy code

SELECT
    cntrsprt.v_nm AS nome_centro_sportivo,
    cntrsprt.v_id AS id_centro_sportivo,
    cnvnzn_cd.dt AS data_inizio_convenzione
FROM
    cntrsprt
LEFT JOIN cnvnzn_cd ON cntrsprt.v_id = cnvnzn_cd.dCntrSprtv
UNION
SELECT
    cntrdnzn_nm AS nome_centro_donazione,
    cntrdnzn_id AS id_centro_donazione,
    cnvnzn_cd.dt AS data_inizio_convenzione
FROM
    cntrdnzn
LEFT JOIN cnvnzn_cd ON cntrdnzn_id = cnvnzn_cd.dCntrDnzn;
```

In questa query, stiamo utilizzando una combinazione di `LEFT JOIN` per unire le tabelle `'cntrsprt'` (centri sportivi) e `'cntrdnzn'` (centri donazione) con la tabella `'cnvnzn_cd'` (convenzioni). L'alias `'cntrsprt'` rappresenta i centri sportivi e l'alias `'cntrdnzn'` rappresenta i centri donazione. La clausola `'UNION'` combina i risultati delle due selezioni in un unico risultato.

CHAT GPT

Database farmacia



Database farmacia

- Caratteristiche:
- Complessità medio-bassa(12 tabelle)
- Presenza di ambiguità (tabelle diverse con attributi con lo stesso nome)
- Non è presente sul Web (quindi è da escludere che il modello lo abbia processato in fase di training)
- Può contenere informazioni sensibili/riservate (ambito medico)
-

Query 1

Livello : 1

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico 3516326614882"

Ecco la query SQL:

sql

Copy

```
SELECT *  
FROM paziente  
WHERE Medico = '3516326614882';
```

Questa query seleziona tutti i record dalla tabella **paziente**, ma vi applica la condizione **WHERE** per filtrare i risultati in modo da restituire solo i pazienti che hanno quel determinato codice medico.

Query 1 - schema ugly

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico 3516326614882"

Ecco la traduzione in SQL:

```
SELECT *
FROM pznt
WHERE Mdc = '3516326614882';
```

Query 1 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico 3516326614882"

Ecco la query in SQL:

sql

Copy

```
SELECT *
FROM pznt
WHERE Mdc = '3516326614882';
```

CLAUDE INSTANT

Query 1

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico
3516326614882"

Per selezionare tutti i pazienti che hanno come medico l'identificativo "3516326614882,"
puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT *
FROM paziente
WHERE Medico = '3516326614882';
```

CHAT GPT

Query 1 - schema ugly

Livello : 1

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico 3516326614882"

Per selezionare tutti i pazienti che hanno come medico il numero "3516326614882" in SQL, puoi utilizzare la seguente query:

sql

 Copy code

```
SELECT *
FROM pazienti
WHERE medico = '3516326614882';
```

CHAT GPT

Query 1 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i pazienti che hanno come medico 3516326614882"

Per selezionare tutti i pazienti che hanno come medico il codice "3516326614882", puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT *
FROM pznt
WHERE Mdc = '3516326614882';
```

CHAT GPT

Query 2

Livello Difficoltà: 1

Esito: Corretto

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

Ecco la query SQL per trovare il codice dei farmaci prodotti dall'azienda 'Bernetti Group':

sql

Copy

```
SELECT Codice  
FROM farmaco  
WHERE Azienda = 'Bernetti Group';
```

Questa query seleziona la colonna **Codice** dalla tabella **farmaco**, ma vi applica la condizione **WHERE** per filtrare i risultati in modo da restituire solo i codici dei farmaci prodotti dall'azienda 'Bernetti Group'.

CLAUDE INSTANT

Query 2 - schema ugly

Livello Difficoltà: 1

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

Ecco la traduzione in SQL della query:

```
SELECT m.Mtrcl AS codice_farmaco  
FROM mnfstn m  
JOIN fftcrltr f ON m.Nm = f.Nm  
WHERE f.Nm = 'Bernetti Group'
```

Query 2 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

sql

Copy

```
SELECT Cdc  
FROM frmc  
WHERE znd = 'Bernetti Group'  
```
```

Questa query seleziona il codice dei farmaci (`Cdc`) dalla tabella `frmc` dove il nome del fornitore (`znd`) è 'Bernetti Group'.

CLAUDE INSTANT

# Query 2

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

Per trovare il codice dei farmaci prodotti dall'azienda "Bernetti Group," puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT Codice
FROM farmaco
WHERE Azienda = 'Bernetti Group';
```

# Query 2 - schema ugly

## Livello Difficoltà: 1

### Esito: Corretto

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

Per trovare il codice dei farmaci prodotti dall'azienda "Bernetti Group," dovresti utilizzare una query SQL con una clausola di selezione (SELECT) per ottenere il codice dei farmaci dalla tabella `frmc` e una clausola di filtro (WHERE) per specificare che l'azienda è "Bernetti Group." Supponendo che il campo contenente il nome dell'azienda sia denominato `Prncpttv` nella tabella `frmc`, la query sarebbe la seguente:

sql

 Copy code

```
SELECT Cdc
FROM frmc
WHERE Prncpttv = 'Bernetti Group';
```

CHAT GPT

# Query 2 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il codice dei farmaci prodotti dall'azienda Bernetti Group"

Per trovare il codice dei farmaci prodotti dall'azienda "Bernetti Group", dovrà utilizzare una query SQL che includa una clausola `WHERE` per filtrare i risultati in base all'azienda di interesse. Assumendo che l'azienda sia identificata nella tabella `firmc` dal campo `znd`, la query SQL sarebbe la seguente:

sql

Copy code

```
SELECT Cdc
FROM firmc
WHERE znd = 'Bernetti Group';
```

**CHAT GPT**

# Query 3

Livello Difficoltà: 1  
Esito: **Corretto \***

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

sql

Copy

```
SELECT PrincipioAttivo
FROM cura
JOIN sintomi ON cura.Sintomo = sintomi.Sintomo
WHERE sintomi.Sintomo = 'mal di testa';
...
```

Questa query effettua un join tra le tabelle `cura` e `sintomi` utilizzando la colonna `Sintomo` in comune.

Quindi applichiamo la condizione `WHERE` per filtrare i risultati e restituire solo i principi attivi che curano il `mal di testa`.

Vengono fatte le seguenti operazioni:

1. Si uniscono le tabelle `cura` e `sintomi` sulla base della colonna `Sintomo` in comune
2. Nella clausola `WHERE` filtriemo i risultati richiedendo solo quelli dove `sintomi.Sintomo` è `mal di testa`
3. Selezioniamo solo la colonna `PrincipioAttivo` dalla tabella `cura`, che ci indica appunto il principio attivo che cura quel sintomo.

In questo modo otterremo come risultato solo i principi attivi che curano il mal di testa.

\*Commento:

# Query 3 - schema ugly

Livello Difficoltà: 1

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

```
SELECT Prncptv
FROM frmc
JOIN ntzzn ON frmc.Cdc = ntzzn.Cdc2
JOIN ffftclliitri ON ffftclliitri.Nm = 'mal di testa'
WHERE ntzzn.ntt = 'cura';
```

# Query 3 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

La query SQL per trovare il principio attivo che cura il mal di testa è la seguente:

sql

Copy

```
SELECT Prncpttv
FROM cr
WHERE Sntm = 'mal di testa'
-- Filtra i record dove il sintomo è 'mal di testa'
```

CLAUDE INSTANT

# Query 3

## Livello Difficoltà: 1

### Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

Per trovare il principio attivo che cura il mal di testa, puoi utilizzare la seguente query SQL:

sql



Copy code

```
SELECT PrincipioAttivo
FROM cura
WHERE Sintomo = 'mal di testa';
```

# Query 3 - schema ugly

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

Per tradurre la frase "Trova il principio attivo che cura il mal di testa" in una query SQL, dovremmo conoscere la struttura delle tabelle coinvolte. Supponendo che ci sia una tabella chiamata "principi\_attivi" con una colonna "nome\_principio" e una tabella chiamata "farmaci" con una colonna "nome\_farmaco" e una colonna "principio\_attivo" che fa riferimento al principio attivo associato a ciascun farmaco, la query potrebbe essere:

sql

 Copy code

```
SELECT nome_principio
FROM principi_attivi
WHERE nome_principio IN (
 SELECT principio_attivo
 FROM farmaci
 WHERE nome_farmaco = 'mal di testa'
);
```

**CHAT GPT**

# Query 3 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova il principio attivo che cura il mal di testa"

Per trovare il principio attivo che cura il mal di testa, dovrà eseguire una query SQL che seleziona il campo "Prncpttv" dalla tabella "cr" (che rappresenta i medicinali) dove il campo "Sntm" è uguale a "mal di testa". Di seguito, ti fornisco la query SQL per ottenere questa informazione:

sql

Copy code

```
SELECT Prncpttv
FROM cr
WHERE Sntm = 'mal di testa';
```

**CHAT GPT**

# Query 4

## Livello : 1

### Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della controindicazione 'atqueodio' "

```
SELECT Descrizione
FROM controindicazioni
WHERE Nome = 'atqueodio';
```

# Query 4 - schema ugly

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della  
controindicazione 'atqueodio' "

```
SELECT Dscrzn
FROM cntrndczn
WHERE Nm = 'atqueodio';
```

# Query 4 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della controindicazione 'atqueodio' "

Seleziona la descrizione della controindicazione 'atqueodio':

sql

Copy

```
SELECT Dscrzn
FROM cntrndczn
WHERE Nm = 'atqueodio';
```

CLAUDE INSTANT

# Query 4

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della controindicazione 'atqueodio' "

Per selezionare la descrizione della controindicazione con nome 'atqueodio', puoi utilizzare la seguente query SQL:

sql



Copy code

```
SELECT Descrizione
FROM controindicazioni
WHERE Nome = 'atqueodio';
```

**CHAT GPT**

# Query 4 - schema ugly

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della controindicazione 'atqueodio' "

La seguente query SQL seleziona la descrizione della controindicazione con il nome 'atqueodio':

sql

Copy code

```
SELECT Dscrzn
FROM cntrndczn
WHERE Nm = 'atqueodio';
```

CHAT GPT

# Query 4 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona la descrizione della controindicazione 'atqueodio' "

Per tradurre la query in SQL, dovresti usare la sintassi del linguaggio SQL per selezionare la descrizione della controindicazione 'atqueodio'. Supponendo che la tabella contenente i dati delle controindicazioni sia chiamata 'cntrndczn' e che il campo che contiene la descrizione sia chiamato 'Dscrzn', la query sarà la seguente:

sql



Copy code

```
SELECT Dscrzn
FROM cntrndczn
WHERE Nm = 'atqueodio';
```

**CHAT GPT**

# Query 5

## Livello : 2

### Esito: Corretto

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

Ecco la query SQL per trovare tutti i pazienti per cui sono stati prescritti farmaci nel 2020:

sql

Copy

```
SELECT p.CodiceFiscale, p.Nome, p.Cognome
FROM paziente p
INNER JOIN ricetta r ON p.CodiceFiscale = r.CodiceFiscale
WHERE YEAR(r.Data) = 2020
```

# Query 5 - schema ugly

Livello : 2

Esito: **Corretto**

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

```
SELECT pznt.Nm, pznt.Cgnm
FROM pznt
JOIN rctt ON pznt.CdcFscl = rctt.CdcFscl
WHERE YEAR(rctt.Dt) = 2020
```

# Query 5 - ugly & insert

Livello : 2

Esito: **Corretto**

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

Ecco la query SQL per trovare tutti i pazienti per cui sono stati prescritti farmaci nel 2020:

sql

Copy

```
SELECT pznt.Nm, pznt.Cgnm
FROM pznt
JOIN rctt ON rctt.CdcFscl = pznt.CdcFscl
WHERE YEAR(rctt.Dt) = 2020
```

CLAUDE INSTANT

# Query 5

## Livello : 2

### Esito: Corretto

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

Per trovare tutti i pazienti per cui sono stati prescritti farmaci nel 2020, è necessario utilizzare la tabella 'ricetta', che contiene le informazioni sulle prescrizioni, inclusa la data di prescrizione. La query SQL per ottenere questi dati sarà la seguente:

sql

 Copy code

```
SELECT DISTINCT p.*
FROM paziente p
JOIN ricetta r ON p.CodiceFiscale = r.CodiceFiscale
WHERE YEAR(r.Data) = 2020;
```

# Query 5 - schema ugly

## Livello : 2

### Esito: **Scorretto**

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

La query SQL per trovare tutti i pazienti per cui sono stati prescritti farmaci nel 2020 dipenderà dalla struttura esatta del database. Supponendo che ci sia una tabella chiamata "prescrizioni" che contiene le informazioni sulle prescrizioni, e un'altra tabella chiamata "pazienti" con le informazioni sui pazienti, la query potrebbe essere scritta come segue:

sql

 Copy code

```
SELECT DISTINCT p.Nome, p.Cognome
FROM pazienti p
INNER JOIN prescrizioni pr ON p.ID_paziente = pr.ID_paziente
WHERE YEAR(pr.Data_prescrizione) = 2020;
```

CHAT GPT

# Query 5 - ugly & insert

Livello : 2

Esito: **Corretto**

Traduci in sql la seguente query : "Trova tutti i pazienti per cui sono stati prescritti farmaci nel 2020"

Per trovare tutti i pazienti per cui sono stati prescritti farmaci nel 2020, puoi utilizzare una query SQL che coinvolga le tabelle `pznt` e `rctt`, dove `rctt` contiene le informazioni sulle prescrizioni di farmaci. Di seguito, trovi la traduzione della query in SQL:

sql

 Copy code

```
SELECT DISTINCT pznt.Nm, pznt.Cgnm
FROM pznt
INNER JOIN rctt ON pznt.CdcFscl = rctt.CdcFscl
WHERE YEAR(rctt.Dt) = 2020;
```

**CHAT GPT**

# Query 6

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalità"

Conta il numero di case farmaceutiche per nazionalità:

```
SELECT Nazionalita, COUNT(*) AS numero
FROM casafarmaceutica
GROUP BY Nazionalita
```

...

# Query 6 - schema ugly

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalità"

```
SELECT nctzndczn.Nm ndczn, COUNT(*) nrFrm
FROM frmc
JOIN pznt ON frmc.Cdc = pznt.Cdc
JOIN cntrndczn ON pznt.Nm = cntrndczn.Nm
JOIN nctzndczn ON cntrndczn.ndczn = nctzndczn.ndczn
GROUP BY nctzndczn.Nm
```

# Query 6 - ugly & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalità"

Ecco la query SQL per contare il numero di case farmaceutiche per nazionalità:

sql

Copy

```
SELECT Nazione, COUNT(*) AS NumeroCase
FROM csfrmctc
GROUP BY Nazione
```

CLAUDE INSTANT

# Query 6

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalità"

Per contare il numero di case farmaceutiche per nazionalità, puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT Nazionalita, COUNT(*) AS NumeroCaseFarmaceutiche
FROM casafarmaceutica
GROUP BY Nazionalita;
```

# Query 6 - schema ugly

## Livello Difficoltà: 2

### Esito: Corretto

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalità"

Per contare il numero di case farmaceutiche per nazionalità, dovrai utilizzare un'operazione di raggruppamento (GROUP BY) in SQL. Supponendo che la tabella che contiene le informazioni sulle case farmaceutiche sia denominata `csfrmctc` e il campo contenente la nazionalità sia denominato `Nznlt`, la query sarà la seguente:

sql



Copy code

```
SELECT Nznlt, COUNT(*) AS NumeroCaseFarmaceutiche
FROM csfrmctc
GROUP BY Nznlt;
```

CHAT GPT

# Query 6 - ugly & insert

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di case farmaceutiche per nazionalita"

Per contare il numero di case farmaceutiche per nazionalità, possiamo utilizzare la funzione di aggregazione `COUNT` e la clausola `GROUP BY`. In questo modo, raggrupperemo le case farmaceutiche in base alla nazionalità e conteremo quanti elementi ci sono in ogni gruppo.

Di seguito la traduzione della query in SQL:

sql

Copy code

```
SELECT Nznlt as Nazionalita, COUNT(*) as NumeroCaseFarmaceutiche
FROM csfrmctc
GROUP BY Nznlt;
```

**CHAT GPT**

# Query 7

## Livello Difficoltà: 2

### Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

sql

Copy

```
SELECT Codice2
FROM interazioni
WHERE Codice1 = (
 SELECT Codice FROM farmaco WHERE NomeCommerciale = 'saepeconsequatur'
);
````
```

Questa query esegue i seguenti passaggi:

1. Seleziona la colonna `Codice2` dalla tabella `interazioni`
2. Vi applica la condizione `WHERE` per filtrare i risultati solo dove `Codice1` è uguale ad una sottoquery
3. La sottoquery seleziona il codice dalla tabella `farmaco` dove il nome commerciale è `saepeconsequatur`

In questo modo otterremo tutti i codici dei farmaci (`Codice2`) che hanno interazioni con il farmaco `saepeconsequatur`.

La tabella `interazioni` infatti contiene i codici dei farmaci che interagiscono tra loro, indicati nelle colonne `Codice1` e `Codice2` .

Il modello identifica correttamente la necessità di risalire al codice dato il nome comune del farmaco ma

- non effettua l'OR, cioè non riconosce la **simmetria** della relazione
- non essendo il NomeCommerciale una chiave, l'utilizzo della query annidata con l'operatore = è errato

SELECT interazioni.*
FROM `interazioni`, farmaco
WHERE (Codice1 = farmaco.Codice **AND**
farmaco.NomeCommerciale = "saepeconsequatur")
OR (Codice2 = farmaco.Codice **AND**
farmaco.NomeCommerciale = "saepeconsequatur")

Query 7 - schema ugly

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

Ecco la query SQL per ottenere tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur':

sql

Copy

```
SELECT Cdc
FROM ntrzn
WHERE Cdc2 IN (
    SELECT Cdc
    FROM frmc
    WHERE NmCmmrc1 = 'saepeconsequatur'
)
```

Query 7 - ugly & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

Ecco la query SQL per ottenere ciò che chiedi:

sql

Copy

```
SELECT c.Cdc
FROM ntrzn n
JOIN frmz f ON n.Cdc1 = f.Cdc
JOIN mnfstn m ON m.Cdc = f.Cdc
WHERE m.Nm = 'saepeconsequatur'
```

Query 7

Livello Difficoltà: 2

Esito: Corretto

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

Per trovare tutti i codici di farmaci che hanno interazioni con il farmaco di nome 'saepeconsequatur', puoi utilizzare una query SQL con una join tra le tabelle `farmaco` e `interazioni` in base ai codici corrispondenti. La query sarebbe la seguente:

sql



Copy code

```
SELECT f.Codice
FROM farmaco f
JOIN interazioni i ON f.Codice = i.Codice1 OR f.Codice = i.Codice2
WHERE f.NomeCommerciale = 'saepeconsequatur';
```

CHAT GPT

Query 7 - schema ugly

Livello Difficoltà: 2

Esito: Corretto

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

Per trovare tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur', puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT DISTINCT m.Cdc
FROM frmrc m
JOIN ntrzn n ON m.Cdc = n.Cdc1 OR m.Cdc = n.Cdc2
WHERE m.NmCmmrc1 = 'saepeconsequatur';
```

CHAT GPT

Query 7 - ugly & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur' "

Per trovare tutti i codici di farmaci che hanno interazioni con il farmaco 'saepeconsequatur', dovremmo eseguire una query SQL che utilizza la tabella `rsch`. La query sarà simile a questa:

sql

 Copy code

```
SELECT DISTINCT Cdc
FROM rsch
WHERE Nm = 'saepeconsequatur' ;
```

CHAT GPT

Query 8

Livello Difficoltà: 2

Esito: Corretta

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico:

```
SELECT s.Sintomo, s.Descrizione  
FROM sintomi s  
JOIN cura c ON s.Sintomo = c.Sintomo  
WHERE c.PrincipioAttivo = 'nome principio attivo'  
ORDER BY s.Sintomo
```

Query 8 - schema ugly

Livello Difficoltà: 2

Esito: Corretta

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

Ecco la query SQL:

sql

Copy

```
SELECT s.Sntm, s.Dscrzn
FROM sntm s
JOIN cr c ON c.Sntm = s.Sntm
WHERE c.Prncpttv = 'principio_attivo_specificato'
ORDER BY s.Sntm
```

Query 8 - ugly & insert

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

```
SELECT s.Sntm, s.Dscrzn  
FROM sntm s  
JOIN cr c ON s.Sntm = c.Sntm  
WHERE c.Prncptv = 'NOME PRINCIPIO ATTIVO'  
ORDER BY s.Sntm
```

Query 8

Livello Difficoltà: 2

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

Per selezionare i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico, puoi utilizzare una query SQL con una join tra le tabelle `cura` e `sintomi`, e un filtro per il principio attivo desiderato. La query sarebbe la seguente:

sql

 Copy code

```
SELECT s.Sintomo, s.Descrizione
FROM cura c
JOIN sintomi s ON c.Sintomo = s.Sintomo
WHERE c.PrincipioAttivo = 'principio_attivo_desiderato'
ORDER BY s.Sintomo ASC;
```

CHAT GPT

Query 8 - schema ugly

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

Per tradurre la query in SQL, assumiamo che il nome della tabella contenente i dati sui sintomi sia `sintomi`, e il nome della tabella contenente i dati sui principi attivi sia `principi_attivi`. Assumiamo inoltre che ci sia una tabella di collegamento chiamata `trattamento` che associa i sintomi ai principi attivi.

La query SQL sarà quindi la seguente:

```
sql Copy code
SELECT s.Sintomo, s.Descrizione
FROM sintomi s
JOIN trattamento t ON s.IdSintomo = t.IdSintomo
JOIN principi_attivi p ON t.IdPrincipioAttivo = p.IdPrincipioAttivo
WHERE p.NomePrincipioAttivo = 'nome_principio_attivo'
ORDER BY s.Sintomo ASC;
```

CHAT GPT

Query 8 - ugly & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona i sintomi e le loro descrizioni che sono curati da un dato principio attivo, ordinati in ordine alfabetico"

La query SQL per selezionare i sintomi e le loro descrizioni curati da un dato principio attivo e ordinarli in ordine alfabetico sarebbe la seguente:

```
sql Copy code
SELECT Sntm, Dscrzn
FROM cr
JOIN sntm ON cr.Sntm = sntm.Sntm
WHERE Prncptv = 'principio_attivo_cercato'
ORDER BY Sntm ASC;
```

CHAT GPT

Query 9

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Ecco la query SQL tradotta:

```
sql Copy
SELECT m.Matricola, m.Nome, m.Cognome, COUNT(*) AS totale_ricette
FROM medico m
JOIN paziente p ON p.Medico = m.Matricola
JOIN ricetta r ON r.CodiceFiscale = p.CodiceFiscale
GROUP BY m.Matricola
ORDER BY totale_ricette DESC
LIMIT 1
```

Query 9 - schema ugly

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Ecco la query SQL per ottenere il medico che ha i pazienti con il numero maggiore di ricette totali:

sql

Copy

```
SELECT m.Mtrcl, m.Nm
FROM mdc m
JOIN pznt p ON p.Mdc = m.Mtrcl
JOIN rctt r ON r.CdcFscl = p.CdcFscl
GROUP BY m.Mtrcl
ORDER BY COUNT(r.dntfctv) DESC
LIMIT 1
```

Query 9 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Ecco la query in SQL per ottenere il medico che ha i pazienti con il maggior numero totale di ricette:

sql

Copy

```
SELECT m.Nm, m.Cgnm, COUNT(*) AS tot_ricette
FROM mdc m
JOIN pznt p ON m.Mtrcl = p.Mdc
JOIN rctt r ON p.CdcFscl = r.CdcFscl
GROUP BY m.Nm, m.Cgnm
ORDER BY tot_ricette DESC
LIMIT 1
```

Query 9

Livello : 3

Esito: Corretto

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Per determinare il medico che ha i pazienti con più ricette totali, puoi utilizzare una query SQL con una combinazione di join, raggruppamento (GROUP BY) e ordinamento (ORDER BY). La query sarà simile a questa:

sql

 Copy code

```
SELECT m.Matricola, m.Nome, m.Cognome, COUNT(r.Identificativo) AS NumeroRicette
FROM medico m
JOIN paziente p ON m.Matricola = p.Medico
LEFT JOIN ricetta r ON p.CodiceFiscale = r.CodiceFiscale
GROUP BY m.Matricola, m.Nome, m.Cognome
ORDER BY NumeroRicette DESC
LIMIT 1;
```

CHAT GPT

Query 9 - schema ugly

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Per tradurre la domanda in SQL, dobbiamo assumere che le tabelle coinvolte siano chiamate "medici", "pazienti" e "ricette", e che ci sia una relazione tra di esse. Inoltre, la domanda richiede di trovare il medico che ha i pazienti con il maggior numero di ricette totali. Dobbiamo quindi utilizzare una query con JOIN e funzioni di aggregazione. Ecco come sarebbe la query SQL:

sql

 Copy code

```
SELECT m.nome_medico, COUNT(r.id_ricetta) AS totale_ricette
FROM medici m
JOIN pazienti p ON m.id_medico = p.id_medico
JOIN ricette r ON p.id_paziente = r.id_paziente
GROUP BY m.nome_medico
ORDER BY totale_ricette DESC
LIMIT 1;
```

CHAT GPT

Query 9 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Quale e' il medico che ha i pazienti con più ricette totali?"

Per determinare il medico che ha i pazienti con più ricette totali, puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT m.Mtrcl, m.Nm, m.Cgnm, COUNT(r.dntfctv) AS Numero_Ricette
FROM mdc m
JOIN pznt p ON m.Mtrcl = p.Mdc
JOIN rctt r ON p.CdcFscl = r.CdcFscl
GROUP BY m.Mtrcl, m.Nm, m.Cgnm
ORDER BY Numero_Ricette DESC
LIMIT 1;
```

CHAT GPT

Query 10

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Ecco la query SQL per selezionare tutti i sintomi curati dalla tachipirina:

sql

Copy

```
SELECT s.*  
FROM sintomi s  
JOIN cura c ON s.Sintomo = c.Sintomo  
WHERE c.PrincipioAttivo = 'Tachipirina'
```

Il modello evita di effettuare alcune join. Tuttavia,
se la tachipirina fosse un principio attivo,
l'approccio scelto sarebbe corretto:
→ ripetiamo fornendo informazione "di dominio"

SELECT sintomi.Sintomo
FROM farmaco, principioattivo, cura, sintomi
WHERE farmaco.NomeCommerciale = 'tachipirina'
AND farmaco.PrincipioAttivo = principioattivo.PrincipioAttivo
AND cura.PrincipioAttivo = principioAttivo.PrincipioAttivo

Query 10 - schema ugly

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Ecco la query SQL:

```
SELECT s.Dscrzn AS sintomo  
FROM sntm s  
INNER JOIN cr c ON c.Sntm = s.Sntm  
WHERE c.Prncpttv = 'Tachipirina'
```

Query 10 - ugly & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Ecco la query in SQL per selezionare tutti i sintomi curati dalla tachipirina:

sql

Copy

```
SELECT s.Sntm AS Sintomo, s.Dscrzn AS Descrizione  
FROM sntm s  
JOIN cr c ON s.Sntm = c.Sntm  
WHERE c.Prncpttv = 'tachipirina'
```

Query 10

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Per selezionare tutti i sintomi curati dalla tachipirina, dovrà utilizzare una query SQL che coinvolga le tabelle `farmaco` e `cura`, in quanto la tabella `cura` collega gli ingredienti attivi (e quindi i farmaci) ai sintomi che trattano.

Assumendo che il principio attivo della tachipirina sia "tachipirina", la query potrebbe essere la seguente:

sql

 Copy code

```
SELECT DISTINCT c.Sintomo
FROM farmaco f
JOIN cura c ON f.PrincipioAttivo = c.PrincipioAttivo
WHERE f.NomeCommerciale = 'tachipirina';
```

CHAT GPT

Query 10 - schema ugly

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Per tradurre la query "Seleziona tutti i sintomi curati dalla tachipirina" in SQL, bisogna conoscere la struttura delle tabelle coinvolte. Tuttavia, dato che non disponiamo di informazioni riguardanti le tabelle specifiche coinvolte, posso fornirti un esempio generico basato su ipotesi.

Supponiamo che ci siano due tabelle coinvolte: una tabella chiamata `farmaci` che contiene informazioni sui farmaci, e un'altra tabella chiamata `sintomi` che contiene informazioni sui sintomi e il farmaco che li cura. Inoltre, ipotizziamo che nella tabella `sintomi`, ci sia una colonna chiamata `farmaco_curante` che indica il nome del farmaco che cura il sintomo.

Con queste ipotesi, la query per selezionare tutti i sintomi curati dalla tachipirina sarebbe la seguente:

sql



Copy code

```
SELECT *
FROM sintomi
WHERE farmaco_curante = 'tachipirina';
```

CHAT GPT

Query 10 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dalla tachipirina"

Per selezionare tutti i sintomi curati dalla tachipirina (paracetamolo) dalla tabella `cr`, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT Sntm  
FROM cr  
WHERE Prncpttv = 'Paracetamolo';
```

CHAT GPT

Query 10.5

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dal farmaco tachipirina"

Ecco la query SQL per selezionare tutti i sintomi curati dal farmaco tachipirina:

sql

Copy

```
SELECT s.*  
FROM sintomi s  
JOIN cura c ON s.Sintomo = c.Sintomo  
JOIN farmaco f ON c.PrincipioAttivo = f.PrincipioAttivo  
WHERE f.NomeCommerciale = 'Tachipirina'
```

Query 10.5 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dal farmaco tachipirina"

Ecco la query SQL:

```
SELECT s.Dscrzn AS sintomo  
FROM sntm s  
INNER JOIN cr c ON c.Sntm = s.Sntm  
INNER JOIN prncpttv p ON p.Prncpttv = c.Prncpttv  
WHERE p.Prncpttv = 'Tachipirina'
```

Query 10.5 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dal farmaco tachipirina"

Seleziona tutti i sintomi curati dal farmaco tachipirina

sql

 Copy

```
SELECT s.Sntm AS sintomo, s.Dscrzn AS descrizione  
FROM sntm s  
INNER JOIN cr c ON s.Sntm = c.Sntm  
WHERE c.Prncpttv = 'tachipirina'
```

Query 10.5 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dal farmaco tachipirina"

Per tradurre la frase in SQL, dobbiamo assumere che ci siano due tavole coinvolte: una tabella chiamata "farmaci" e un'altra chiamata "sintomi". Supponiamo che nella tabella "farmaci" ci sia una colonna chiamata "nome_farmaco" e nella tabella "sintomi" ci sia una colonna chiamata "sintomo_curato".

La query SQL per selezionare tutti i sintomi curati dal farmaco "tachipirina" sarebbe la seguente:

sql

 Copy code

```
SELECT sintomo_curato  
FROM sintomi  
WHERE nome_farmaco = 'tachipirina';
```

CHAT GPT

Query 10.5 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona tutti i sintomi curati dal farmaco tachipirina"

Per selezionare tutti i sintomi curati dal farmaco "tachipirina" dalla tabella `cr`, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT Sntm  
FROM cr  
WHERE Prncpttv = 'tachipirina';
```

CHAT GPT

Query 11

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Dato il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

```
SELECT r.Codice, f.NomeCommerciale, r.Data  
FROM ricetta r  
JOIN farmaco f ON f.Codice = r.Codice  
JOIN medico m ON m.Matricola = r.Matricola  
JOIN paziente p ON p.CodiceFiscale = r.CodiceFiscale  
WHERE p.CodiceFiscale = 'codice_fiscale'  
AND m.Nome = 'Tiziano' AND m.Cognome = 'Bova'  
ORDER BY r.Data
```

Query 11 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Dato il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

Questa è la traduzione in SQL della query:

```
SELECT rctt.*  
FROM rctt  
JOIN pznt ON rctt.CdcFscl = pznt.CdcFscl  
JOIN mdc ON pznt.Mdc = mdc.Mtrcl  
WHERE mdc.Nm = 'Tiziano Bova'  
AND rctt.CdcFscl = 'CodiceFiscalePaziente'  
ORDER BY rctt.Dt;
```

Query 11 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Dato il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

```
SELECT f.NmCmmrc1 AS Farmaco, r.Dt AS Data  
FROM rctt r  
JOIN frmc f ON r.Cdc = f.Cdc  
JOIN mdc m ON r.Mtrcl = m.Mtrcl  
WHERE r.CdcFscl = :CodiceFiscalePaziente AND m.Nm = 'Tiziano'  
AND m.Cgnm = 'Bova'  
ORDER BY r.Dt
```

Query 11

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Data il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

Per ottenere tutti i farmaci che il dottor Tiziano Bova ha prescritto a un paziente specifico, dato il suo codice fiscale, dovremo coinvolgere le tabelle `paziente`, `medico`, `ricetta`, e `farmaco`. Queste tabelle contengono le informazioni necessarie per trovare i farmaci prescritti da un medico a un paziente specifico.

Assumiamo che il campo `CodiceFiscale` della tabella `paziente` identifichi univocamente il paziente e che il campo `Matricola` della tabella `medico` identifichi univocamente il medico Tiziano Bova.

La query SQL sarebbe la seguente:

```
sql Copy code
SELECT f.NomeCommerciale, r.Data
FROM paziente p
JOIN medico m ON p.Medico = m.Matricola
JOIN ricetta r ON p.CodiceFiscale = r.CodiceFiscale
JOIN farmaco f ON r.Codice = f.Codice
WHERE m.Nome = 'Tiziano' AND m.Cognome = 'Bova'
    AND p.CodiceFiscale = 'CODICE_FISCALE_DEL_PAZIENTE'
ORDER BY r.Data;
```

CHAT GPT

Query 11 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Dato il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

La query SQL per trovare tutti i farmaci che il dottor Tiziano Bova ha prescritto a un paziente, dato il codice fiscale del paziente e ordinati per data, potrebbe essere la seguente:

```
sql Copy code
SELECT f.NmCmmrcI, r.Dt
FROM frmC f
JOIN rctt r ON f.Cdc = r.Cdc
JOIN cr c ON r.CdcFscl = c.Prncptv
JOIN sntm s ON c.Sntm = s.Sntm
WHERE s.Sntm = 'Tiziano Bova' AND r.CdcFscl = 'CODICE_FISCALE_PAZIENTE'
ORDER BY r.Dt;
```

Query 11 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Dato il codice fiscale di un paziente, trova tutti i farmaci che il dottor Tiziano Bova gli ha prescritto, ordinati per data"

La query SQL per trovare tutti i farmaci prescritti dal dottor Tiziano Bova a un paziente, dato il suo codice fiscale, ordinati per data, può essere scritta nel seguente modo:

```
sql Copy code
SELECT rctt.dntfctv, frmrc.NmCmmrcl, prncpttv.Prncpttv
FROM rctt
JOIN pznt ON rctt.CdcFscl = pznt.CdcFscl
JOIN mdc ON pznt.Mdc = mdc.Mtrcl
JOIN frmrc ON rctt.Cdc = frmrc.Cdc
JOIN prncpttv ON frmrc.Prncpttv = prncpttv.Prncpttv
WHERE mdc.Nm = 'Tiziano' AND mdc.Cgnm = 'Bova' AND pznt.CdcFscl = 'CODICE_FI
ORDER BY rctt.Dt;
```

CHAT GPT

Query 12

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Quale e' la media di controindicazioni per ogni farmaco?"

Ecco la query SQL per ottenere la media di controindicazioni per ogni farmaco:

sql

Copy

```
SELECT f.NomeCommerciale, COUNT(r.Nome) / COUNT(DISTINCT f.Codice)
AS media_controindicazioni
FROM farmaco f
JOIN rischio r ON r.Codice = f.Codice
GROUP BY f.NomeCommerciale
```

Query 12 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Quale e' la media di controndicazioni per ogni farmaco?"

Ecco la query SQL per calcolare la media di controndicazioni per ogni farmaco:

```
SELECT f.NmCmmrcl, COUNT(r.Nm)/COUNT(DISTINCT f.Cdc) AS  
avg_controndicazioni  
FROM cntrndczn r  
JOIN mnfstn m ON m.Nm = r.Nm  
JOIN frmc f ON f.Cdc = m.Cdc  
GROUP BY f.NmCmmrcl
```

Query 12 - ugly & insert

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Quale e' la media di controindicazioni per ogni farmaco?"

Ecco la query SQL per ottenere la media di controindicazioni per ogni farmaco:

sql

Copy

```
SELECT frmc.Prncpttv, AVG(rsch.Frqnz) AS media_controindicazioni
FROM frmc
LEFT JOIN rsch ON frmc.Cdc = rsch.Cdc
GROUP BY frmc.Prncpttv
```

Query 12

Livello Difficoltà: 3

Esito: Corretto

Traduci in SQL la seguente query: "Quale e' la media di controindicazioni per ogni farmaco?"

Per calcolare la media delle controindicazioni per ogni farmaco, dovremo utilizzare una query SQL che coinvolge la tabella "farmaco" e la tabella "rischio" e calcola il numero medio di contraindications (controindicazioni) per ciascun farmaco. Supponendo che la tabella "rischio" abbia un campo "Codice" per identificare il farmaco a cui si riferisce la controindicazione, possiamo scrivere la query in questo modo:

```
sql Copy code
SELECT
    f.Codice AS CodiceFarmaco,
    f.NomeCommerciale AS NomeFarmaco,
    COUNT(r.Nome) AS NumeroControindicazioni,
    AVG(COUNT(r.Nome)) AS MediaControindicazioni
FROM
    farmaco f
LEFT JOIN
    rischio r ON f.Codice = r.Codice
GROUP BY
    f.Codice, f.NomeCommerciale;
```

CHAT GPT

Query 12 - ugly schema

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Quale e' la media di contraddizioni per ogni farmaco?"

Per calcolare la media delle contraddizioni per ogni farmaco, dovrà utilizzare una query SQL con l'aggregazione dei dati. Supponendo che i dati siano memorizzati nella tabella "frmcl" e che il numero di contraddizioni sia rappresentato dalla colonna "contraddizioni" nella stessa tabella, ecco come sarebbe la query SQL:

sql



Copy code

```
SELECT NmCmmrc1 AS Farmaco, AVG(contraddizioni) AS MediaContraddizioni  
FROM frmcl  
GROUP BY NmCmmrc1;
```

CHAT GPT

Query 12 - ugly & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Quale e' la media di contraddizioni per ogni farmaco?"

La traduzione della query in SQL per calcolare la media di contraddizioni per ogni farmaco sarebbe la seguente:

sql

 Copy code

```
SELECT frmcl.NmCmmrc1 AS Farmaco,
       COUNT(ntrzn.Cdc2) AS NumeroContraddizioni,
       AVG(ntrzn.Cdc2) AS MediaContraddizioni
  FROM frmcl
 LEFT JOIN ntrzn ON frmcl.Cdc = ntrzn.Cdc1
 GROUP BY frmcl.NmCmmrc1;
```

CHAT GPT

Query

Livello Difficoltà: 4

Esito: Corretto

...

Traduci in SQL la seguente query: "Seleziona le aziende che producono farmaci che hanno un numero di effetti collaterali sopra la media"

sql

Copy

```
SELECT c.Azienda
FROM casafarmaceutica c
JOIN farmaco f ON c.Azienda = f.Azienda
JOIN manifestano m ON f.Codice = m.Codice
GROUP BY c.Azienda
HAVING COUNT(DISTINCT m.Nome) > (
    SELECT AVG(count)
    FROM (
        SELECT COUNT( DISTINCT m.Nome) AS count
        FROM manifestano m
        JOIN farmaco f ON m.Codice = f.Codice
        GROUP BY f.Codice
    ) c
)
```

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona le aziende che producono farmaci che hanno un numero di effetti collaterali sopra la media"

Ecco la query SQL tradotta:

```
sql Copy
SELECT znd
FROM frmc
JOIN mnfstn ON frmc.Cdc = mnfstn.Cdc
GROUP BY znd
HAVING COUNT(mnfstn.Nm) > (
    SELECT AVG(c)
    FROM (
        SELECT COUNT(mnfstn.Nm) AS c
        FROM frmc
        JOIN mnfstn ON frmc.Cdc = mnfstn.Cdc
        GROUP BY frmc.Cdc
    ) t
)
```

Query - ugly schema

Livello Difficoltà: 4

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona le aziende che producono farmaci che hanno un numero di effetti collaterali sopra la media"

Per selezionare le aziende che producono farmaci con un numero di effetti collaterali sopra la media, dovresti utilizzare una query SQL che calcoli la media del numero di effetti collaterali per ciascuna azienda e quindi selezioni solo le aziende che hanno un numero di effetti collaterali superiore a questa media. Supponendo che tu abbia le seguenti tabelle:

Tabella "aziende":

- NomeAzienda (nome dell'azienda)
- CodiceAzienda (codice identificativo dell'azienda)

Tabella "farmaci":

- NomeFarmaco (nome del farmaco)
- CodiceAzienda (codice identificativo dell'azienda che produce il farmaco)
- NumeroEffettiCollaterali (numero di effetti collaterali del farmaco)

La query in SQL sarebbe la seguente:

```
sql Copy code  
SELECT a.NomeAzienda  
FROM aziende a  
JOIN farmaci f ON a.CodiceAzienda = f.CodiceAzienda  
WHERE f.NumeroEffettiCollaterali > (  
    SELECT AVG(NumeroEffettiCollaterali) AS MediaEffettiCollaterali  
    FROM farmaci  
    GROUP BY CodiceAzienda  
)
```

CHAT GPT

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona le aziende che producono farmaci che hanno un numero di effetti collaterali sopra la media"

Per tradurre la query in SQL, dovrà utilizzare una subquery per calcolare la media dei farmaci e successivamente effettuare una selezione per trovare le aziende che producono farmaci con un numero di effetti collaterali sopra questa media. Di seguito, troverai la query tradotta:

```
sql Copy code
SELECT c.znd
FROM csfrmctc c
WHERE (
    SELECT COUNT(*)
    FROM frmctc f
    WHERE f.znd = c.znd
) > (
    SELECT AVG(effetti_collaterali)
    FROM (
        SELECT znd, COUNT(*) as effetti_collaterali
        FROM frmctc
        GROUP BY znd
    ) AS subquery
);
```

CHAT GPT

Query

Livello Difficoltà: 4

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova I medici che prescrivono il minimo numero di farmaci di derivazione naurale"

```
SELECT m.Nome, m.Cognome, COUNT(f.PrincipioAttivo) AS  
NumeroFarmaci  
FROM medico m  
LEFT JOIN ricetta r ON m.Matricola = r.Matricola  
LEFT JOIN farmaco f ON r.Codice = f.Codice  
LEFT JOIN principioattivo pa ON f.PrincipioAttivo =  
pa.PrincipioAttivo  
WHERE pa.Derivazione = 'naturale'  
GROUP BY m.Matricola  
ORDER BY NumeroFarmaci ASC  
LIMIT 1
```

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo di ricette che ha un paziente"

```
SELECT m.*  
FROM mdct m  
JOIN pznt p ON p.Mdc = m.Mtrcl  
JOIN rctt r ON p.CdcFscl = r.CdcFscl  
WHERE r.Cdc IN (  
    SELECT Cdc  
    FROM rctt  
    GROUP BY CdcFscl  
    HAVING COUNT() > (  
        SELECT MIN(COUNT())  
        FROM rctt  
        GROUP BY CdcFscl  
    )  
)
```

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova i medici che prescrivono il minimo numero di farmaci di derivazione aurale"

Ecco la query tradotta in SQL:

```
sql Copy
SELECT m.Nm, m.Cgnm
FROM mdc m
JOIN pznt p ON m.Mtrcl = p.Mdc
JOIN rctt r ON p.CdcFscl = r.CdcFscl
JOIN frmfc f ON r.Cdc = f.Cdc
JOIN prncpttv pr ON f.Prncpttv = pr.Prncpttv
WHERE pr.Drvzn = 'Naturale'
GROUP BY m.Mtrcl
HAVING COUNT(DISTINCT f.Prncpttv) = (
    SELECT MIN(cnt)
    FROM (
        SELECT COUNT(DISTINCT f.Prncpttv) cnt
        FROM mdc m
        JOIN pznt p ON m.Mtrcl = p.Mdc
        JOIN rctt r ON p.CdcFscl = r.CdcFscl
        JOIN frmfc f ON r.Cdc = f.Cdc
        JOIN prncpttv pr ON f.Prncpttv = pr.Prncpttv
        WHERE pr.Drvzn = 'Naturale'
        GROUP BY m.Mtrcl
    ) t1
)
```

Query - ugly schema

Livello Difficoltà: 4

Esito: **Scorretto**

Traduci in SQL la seguente query: "Trova i medici che prescrivono il minimo numero di farmaci di derivazione naurale"

Per tradurre la query in SQL, dovremo assumere che il database abbia una struttura corretta con le tabelle pertinenti. Poiché non dispongo delle informazioni complete sulle tabelle coinvolte, ti fornirò una query generica che dovrebbe darti il risultato desiderato.

Assumeremo che la tabella dei medici sia chiamata "medici" e che la tabella dei farmaci sia chiamata "farmaci".

```
sql Copy code
SELECT m.NomeMedico, COUNT(p.IdFarmaco) AS NumeroFarmaciPrescritti
FROM medici m
JOIN prescrizioni p ON m.IdMedico = p.IdMedico
JOIN farmaci f ON p.IdFarmaco = f.IdFarmaco
WHERE f.Derivazione = 'Naturale'
GROUP BY m.NomeMedico
HAVING COUNT(p.IdFarmaco) = (
    SELECT MIN(NumFarmaciPrescritti)
    FROM (
        SELECT m2.IdMedico, COUNT(p2.IdFarmaco) AS NumFarmaciPrescritti
        FROM medici m2
        JOIN prescrizioni p2 ON m2.IdMedico = p2.IdMedico
        JOIN farmaci f2 ON p2.IdFarmaco = f2.IdFarmaco
        WHERE f2.Derivazione = 'Naturale'
        GROUP BY m2.IdMedico
    ) AS MinNumFarmaci
);
```

CHAT GPT

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Trova i medici che prescrivono il minimo numero di farmaci di derivazione naturale"

La seguente query SQL trova i medici che prescrivono il minimo numero di farmaci di derivazione naturale:

```
sql Copy code
SELECT mdc.Nm, COUNT(DISTINCT rctt.Mtrcl) AS NumeroFarmaciNaturaliPrescritti
FROM mdc
LEFT JOIN rctt ON mdc.Mtrcl = rctt.Mtrcl
LEFT JOIN prncpttv ON rctt.Cdc = prncpttv.Prncpttv
WHERE prncpttv.Drvzn = 'Naturale'
GROUP BY mdc.Nm
HAVING COUNT(DISTINCT rctt.Mtrcl) = (
    SELECT COUNT(DISTINCT rctt.Mtrcl) AS NumeroFarmaciNaturaliPrescritti
    FROM mdc
    LEFT JOIN rctt ON mdc.Mtrcl = rctt.Mtrcl
    LEFT JOIN prncpttv ON rctt.Cdc = prncpttv.Prncpttv
    WHERE prncpttv.Drvzn = 'Naturale'
    GROUP BY mdc.Nm
    ORDER BY COUNT(DISTINCT rctt.Mtrcl)
    LIMIT 1
);
```

CHAT GPT

Query

Livello Difficoltà: 4

Esito: **Scorretto**

Traduci in SQL la seguente query: "Seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo raggruppando per medico"

Seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo raggruppando per medico

sql

 Copy

```
SELECT m.Matricola, m.Nome, m.Cognome
FROM medico m
JOIN paziente p ON m.Matricola = p.Medico
JOIN ricetta r ON p.CodiceFiscale = r.CodiceFiscale
GROUP BY m.Matricola
HAVING COUNT(r.Identificativo) > (
    SELECT MIN(COUNT(Identificativo)) FROM ricetta GROUP BY CodiceF
iscale
)
```

Query - ugly schema

Livello Difficoltà: 4

Esito: Corretto

Traduci in SQL la seguente query: "Seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo di ricette che ha un paziente"

La seguente query SQL seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo di ricette che ha un paziente:

```
sql Copy code
SELECT DISTINCT m.Nm
FROM mdc m
JOIN pznt p ON m.Mtrcl = p.Mdc
JOIN rctt r ON p.CdcFscl = r.CdcFscl
WHERE r.Cdc IN (
    SELECT r1.Cdc
    FROM rctt r1
    JOIN pznt p1 ON r1.CdcFscl = p1.CdcFscl
    GROUP BY r1.Cdc
    HAVING COUNT(r1.dntfctv) > (
        SELECT COUNT(r2.dntfctv)
        FROM rctt r2
        JOIN pznt p2 ON r2.CdcFscl = p2.CdcFscl
        GROUP BY r2.Cdc
        ORDER BY COUNT(r2.dntfctv) ASC
        LIMIT 1
    )
);
```

CHAT GPT

Query - ugly & insert

Livello : 4

Esito: **Corretto**

Traduci in SQL la seguente query: "Seleziona i medici che hanno pazienti che hanno un numero di ricette maggiore del valore minimo di ricette che ha un paziente"

Per tradurre la query in SQL, possiamo utilizzare una sottoquery per calcolare il valore minimo di ricette che ha un paziente e quindi selezionare i medici che hanno pazienti con un numero di ricette maggiore di questo valore minimo. Ecco come appare la query SQL:

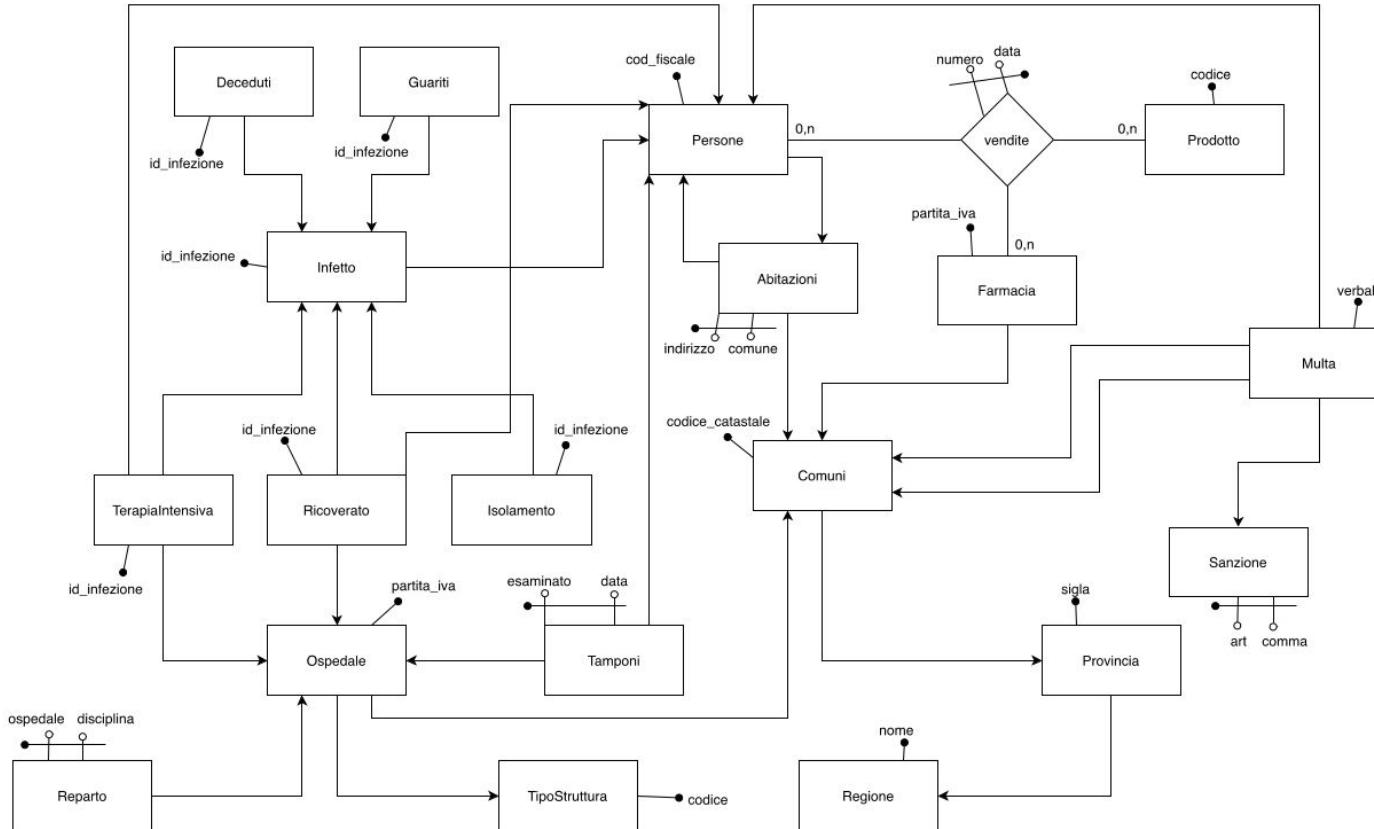
```
sql Copy code
SELECT DISTINCT M.Nm, M.Cgnm
FROM mdc M
JOIN rctt R ON M.Mtrcl = R.Mtrcl
WHERE R.Cdc IN (
    SELECT Cdc
    FROM rctt
    GROUP BY Cdc
    HAVING COUNT(*) > (
        SELECT COUNT(*) AS min_count
        FROM rctt
        GROUP BY Cdc
        ORDER BY min_count
        LIMIT 1
    )
);
```

CHAT GPT

Database Covid

Coronavirus

Database di infetti italiani



Database Covid

Caratteristiche:

- Complessità medio-bassa(20 tabelle)
- Presenza di ambiguità (tabelle diverse con attributi con lo stesso nome)
- Attributi con nome che non descrivono chiaramente il contenuto (nas_comune,res_comune rispettivamente indicano comune di nascita e di residenza)
- Non è presente sul Web (quindi è da escludere che il modello lo abbia processato in fase di training)
- Può contenere informazioni sensibili/riservate (recensioni,visite,donazioni...)

Query 1

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

Ecco la query SQL per calcolare il numero di guariti in una certa data:

sql

Copy

```
SELECT COUNT(*) AS num_guariti  
FROM guariti  
WHERE data_guarigione = '2020-05-15';
```

Query 1 - ugly schema

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

Ecco la query SQL per calcolare il numero di guariti in una certa data:

sql

Copy

```
SELECT COUNT(*)
FROM grt
WHERE dt_grgn = :data;
```

Query 1 - ugly & insert

Livello Difficoltà: 1

Esito: **Scorretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

```
SELECT COUNT(*)  
FROM nftt  
WHERE dt_nfzn = '2020-05-01' AND stt = 'guarito';
```

Query 2

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

Ecco la query SQL per calcolare il numero di farmacie aperte nel 2020:

sql

Copy

```
SELECT COUNT(*) AS num_farmacie  
FROM farmacie  
WHERE YEAR(inizio_attivita) = 2020;
```

Query 2 - ugly schema

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

Seleziona FROM frmcdove frm c sta per farmacie e hanno aperto nel 2020 significa dove la data di apertura è nel 2020. Quindi la query in SQL sarebbe:

sql

Copy

```
SELECT COUNT(*)  
FROM frm  
WHERE nz_ttvtt >= '2020-01-01' AND nz_ttvtt < '2021-01-01'
```

CLAUDE INSTANT

Query 2 - ugly & insert

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

```
SELECT COUNT(*)
FROM frmc
WHERE YEAR(nz_ttv) = 2020
```

La query farà:

- Una selezione nella tabella frmc che contiene le informazioni sulle farmacie
- Utilizza la funzione YEAR() per estrarre solo l'anno di apertura (colonna nz_ttv)
- Il filtro WHERE YEAR(nz_ttv) = 2020 seleziona solo le farmacie aperte nel 2020
- La funzione COUNT() conta il numero di righe soddisfacenti, ovvero il numero di farmacie aperte nel 2020

Ad esempio, se abbiamo questi dati:

```
prtt_v, nz_ttv
100018180, 2019-03-01
100555183, 2019-05-01
101401020, 2020-07-01
101733030, 2020-06-01
```

La query ci restituirà:

```
COUNT(*) : 2
```

In quanto solo gli ultimi due record hanno un anno di apertura (es: YEAR(nz_ttv)) pari al 2020.

CLAUDE INSTANT

Query 3

Livello : 1

Esito: **Corretto**

...

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

Ecco la query SQL per calcolare il totale degli asintomatici nel mese di luglio 2020:

sql

Copy

```
SELECT COUNT(*) AS totale_asintomatici
FROM infetti
WHERE MONTH(data_infezione) = 7
    AND YEAR(data_infezione) = 2020
    AND stato = 'asintomatico';
```

Query 3 - ugly schema

Livello Difficoltà: 1

Esito: **Corretto**

...

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

Ecco la traduzione in SQL della query:

```
SELECT COUNT(*) AS totale_asintomatici  
FROM nfft  
JOIN dc当地 ON nfft.d_nfzn = dc当地.d_nfzn  
WHERE MONTH(dc当地.dt_dcss) = 7  
AND YEAR(dc当地.dt_dcss) = 2020  
AND nfft.stt = 'asintomatico';
```

Query 3 - ugly & insert

Livello Difficoltà: 1

Esito: **Corretto**

...

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

```
SELECT COUNT(*)  
FROM nftt  
WHERE MONTH(dt_nfzn) = 7  
AND YEAR(dt_nfzn) = 2020  
AND stt = 'asintomatico';
```

Query 4

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte"

Per tradurre in SQL la query richiesta, si può usare la seguente istruzione:

sql

Copy

```
SELECT *
FROM multe
WHERE comune_provenienza = 'AAAA' AND data = '2020-01-01'
```

Query 4 - ugly schema

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in sql la seguente query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte"

La query SQL per ottenere tutte le multe fatte in un determinato comune per una data specificata è la seguente:

```
SELECT *
FROM mlt
WHERE cmn_prvnnz = 'XXXX' AND dt = '2020-01-01'
```

Query 4 - ugly & insert

Livello Difficoltà: 1

Esito: **Corretto**

Traduci in sql la seguente query "Dato un comune ed una data,
seleziona tutte le multe che sono state fatte"

```
SELECT *  
FROM mlt  
WHERE cmn_prvnnz = 'XXX' AND dt >= 'YYYY-MM-DD'
```

Query 5

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Selezione il numero di abitanti per ogni comune".

```
SELECT comune, COUNT(*) AS abitanti  
FROM persone  
GROUP BY comune
```

La query seleziona:

- il campo comune (ovvero il comune di residenza della persona)
- il COUNT(*), che calcola il numero di righe totali per ogni comune
- infine applica un GROUP BY comune, in modo da raggruppare i risultati per singolo comune

Il risultato sarà una tabella con:

- il codice catastale del comune
- il numero di residenti (abitanti) nel comune

*la colonna comune non esiste.
Andava considerata la colonna "res_comune" (contiene il comune di residenza), aggiungendo "comune di residenza" funziona

CLAUDE INSTANT

Query 5 - ugly schema

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per ogni comune".

Ecco la query SQL per selezionare il numero di abitanti per ogni comune:

```
SELECT cmn.cdc_ctstl AS codice_comune, btzn.mq AS  
numero_abitanti  
FROM cmn  
INNER JOIN btzn ON cmn.cdc_ctstl = btzn.cmn
```

Query 5 - ugly & insert

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per ogni comune".

Purtroppo dallo schema SQL fornito non è possibile calcolare il numero di abitanti per comune in modo preciso.

Non è presente una tabella che associa i codici dei comuni (cmn.cdc_ctstl) alla loro popolazione.

Per calcolare il numero di abitanti per comune in modo approssimativo, potremmo contare il numero totale di record per ogni codice comune, assumendo che ogni record corrisponda ad un abitante, come nell'esempio:

```
SELECT cmc.cdc_ctstl AS 'Codice Comune', cmc.nm AS 'Comune',  
COUNT(*) AS 'Numero Abitanti'  
FROM cmn AS cmc  
LEFT JOIN btzn ON cmc.cdc_ctstl = btzn.cmn  
GROUP BY cmc.cdc_ctstl
```

CLAUDE INSTANT

Query 5.5 - ugly schema

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per comune di residenza".

Ecco la query SQL per selezionare il numero di abitanti per comune di residenza:

sql

Copy

```
SELECT cmn.nm AS comune, COUNT(*) AS numero_abitanti
FROM prsn
INNER JOIN btzn ON prsn.ndrzz = btzn.ndrzz AND prsn.rs_cmn = btzn.c
mn
INNER JOIN cmn ON btzn.cmn = cmn.cdc_ctstl
GROUP BY cmn.nm
```

Query 6

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

Ecco la traduzione in SQL della query:

sql

Copy

```
SELECT
    p.codice,
    p.nome,
    SUM(v.quantita) AS quantita_totale,
    SUM(v.totale) AS ricavo_totale
FROM prodotti p
JOIN vendite v ON v.prodotto = p.codice
GROUP BY p.codice, p.nome
ORDER BY quantita_totale DESC;
```

CLAUDE INSTANT

Query 6 - ugly schema

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

Ecco la traduzione in SQL della query:

```
SELECT prdtt,  
SUM(qntt) AS qntt_tot,  
SUM(ttl) AS ricavo_tot  
FROM vndt  
GROUP BY prdtt  
ORDER BY qntt_tot DESC
```

Query 6 - ugly & insert

Livello Difficoltà: 2

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

```
SELECT prdtt, SUM(qntt) AS tot_qnt, SUM(ttl) AS tot_ricavo  
FROM vndt  
GROUP BY prdtt  
ORDER BY tot_qnt DESC;
```

Query 7

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Ecco la query SQL per ottenere il numero di infermieri che si sono infettati:

sql

Copy

```
SELECT COUNT(*)  
FROM persone  
INNER JOIN infetti ON persone.cod_fiscale = infetti.persona  
WHERE professione LIKE 'infermiere%'
```

Query 7 - ugly schema

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Ecco la query SQL per ottenere il numero di infermieri infetti:

```
SELECT COUNT(*)  
FROM nftt  
WHERE stt = 'positivo'  
AND prsn IN (  
    SELECT cd_fscl  
    FROM prsn  
    WHERE prfssn = 'Infermiere'  
)
```

Query 7 - ugly & insert

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati".

Ecco la query SQL per ottenere il numero di infermieri infetti:

```
sql
SELECT COUNT(*) AS numero_infermieri_infetti
FROM prsn
WHERE prfssn = 'Infermiere'
AND cd_fscl IN (
    SELECT prsn
    FROM nftt
    WHERE stt IN ('asintomatico','lieve','grave','critico')
)
```

Query 8

Livello : 2

Esito: **Corretto***

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Ecco la query SQL per contare il numero di ospedali per provincia e ritornare la lista ordinata:

```
sql
SELECT
    p.nome, p.sigla, COUNT(*) AS numero_ospedali
FROM
    ospedali o
    JOIN comuni c ON c.codice_catastale = o.comune
    JOIN province p ON p.sigla = c.provincia
GROUP BY
    p.sigla, p.nome
ORDER BY
    numero_ospedali DESC
```

si puo usare una tabella in meno se non si e' interessati al nome della provincia

CLAUDE INSTANT

Query 8 - ugly schema

Livello Difficoltà: 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Ecco la query SQL per contare il numero di ospedali per provincia e ordinare i risultati:

sql

Copy

```
SELECT prvnc.nm AS provincia, COUNT(*) AS numero_ospedali
FROM Frm c
INNER JOIN cmn ON cmn.cdc_ctstl = frm c.cmn
INNER JOIN prvnc ON prvnc.sgl = cmn.prvnc
GROUP BY prvnc.nm
ORDER BY numero_ospedali DESC
```

Query 8 - ugly & insert

Livello Difficoltà: 2

Esito: **Corretto**

si puo usare una tabella in meno se non si e' interessati al nome della provincia

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Ecco la query SQL per ottenere il numero di ospedali per provincia ordinato:

sql

Copy

```
SELECT prvnc.sgl, prvnc.nm, COUNT(*) AS numero_ospedali
FROM spdl
JOIN cmn ON cmn.cdc_ctstl = spdl.cmn
JOIN prvnc ON prvnc.sgl = cmn.prvnc
GROUP BY prvnc.sgl, prvnc.nm
ORDER BY numero_ospedali DESC
```

Query 8.5

Livello : 2

Esito: **Corretto***

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata, mostrando solo la sigla della provincia e il conteggio"

Ecco la query SQL:

```
SELECT p.sigla, COUNT(*) AS numero_ospedali  
FROM ospedali o  
JOIN comuni c ON o.comune = c.codice_catastale  
JOIN province p ON c.provincia = p.sigla  
GROUP BY p.sigla  
ORDER BY numero_ospedali DESC
```

Query 9

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

Ecco la query SQL per ottenere i dati richiesti per ogni regione:

```
sql Copy
SELECT
    r.nome AS regione,
    COUNT(g.id_infezione) AS guariti,
    COUNT(d.id_infezione) AS decessi,
    COUNT(i.id_infezione) - COUNT(g.id_infezione) - COUNT(d.id_infezione) AS positivi,
    COUNT(i.id_infezione) AS totale
FROM
    regioni r
    LEFT JOIN persone p ON r.nome = p.res_region
    LEFT JOIN infetti i ON p.cod_fiscale = i.persona
    LEFT JOIN guariti g ON i.id_infezione = g.id_infezione
    LEFT JOIN deceduti d ON i.id_infezione = d.id_infezione
GROUP BY
    r.nome
```

res_region
e'
campo non
presente

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

```
Ecco la query in SQL:
sql Copy
SELECT
    r.nome AS regione,
    COUNT(g.id_infezione) AS guariti,
    COUNT(d.id_infezione) AS decessi,
    COUNT(i.id_infezione) - COUNT(g.id_infezione) - COUNT(d.id_infezione) AS positivi,
    COUNT(i.id_infezione) AS totale
FROM
    regioni r
    LEFT JOIN persone p ON r.nome = p.res_region
    LEFT JOIN infetti i ON p.cod_fiscale = i.persona
    LEFT JOIN guariti g ON i.id_infezione = g.id_infezione
    LEFT JOIN deceduti d ON i.id_infezione = d.id_infezione
GROUP BY
    r.nome
```

CLAUDE INSTANT

Query 9 - ugly schema

Livello Difficoltà: 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

Ecco la query SQL tradotta:

```
sql Copy
SELECT
    r.nm AS regione,
    COUNT(IF(n.stt = 'guarito', 1, NULL)) AS guariti,
    COUNT(IF(n.stt = 'grave' OR n.stt = 'critico', 1, NULL)) AS decessi,
    COUNT(IF(n.stt = 'lieve' OR n.stt = 'asintomatico', 1, NULL)) AS positivi,
    COUNT(*) AS totale
FROM nftt n
JOIN prsn p ON n.prsn = p.cd_fscl
JOIN btzn b ON p.ndrzz = b.ndrzz AND p.rs_cmn = b.cmn
JOIN cmn c ON b.cmn = c.cdc_ctstl
JOIN prvnc v ON c.prvnc = v.sgl
JOIN rgn r ON v.rgn = r.nm
GROUP BY r.nm
```

Query 9 - ugly & insert

Livello Difficoltà: 2

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

Ecco la query SQL tradotta:

sql

Copy

```
SELECT
    rgn.nm AS regione,
    COUNT(CASE WHEN nftt.stt = 'guarito' THEN 1 END) AS guariti,
    COUNT(CASE WHEN nftt.stt = 'deceduto' THEN 1 END) AS decessi,
    COUNT(CASE WHEN nftt.stt = 'positivo' THEN 1 END) AS attualmente_positivi,
    COUNT(*) AS casi_totali
FROM nftt
INNER JOIN prsn ON nftt.prsn = prsn.cd_fscl
INNER JOIN cmn ON prsn.ns_cmn = cmn.cdc_ctstl
INNER JOIN prvnc ON cmn.prvnc = prvnc.sg1
INNER JOIN rgn ON prvnc.rgn = rgn.nm
GROUP BY rgn.nm;
```

Query 10

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Ecco la query SQL per ottenere il numero di pazienti ricoverati affidati a ciascun infermiere o anestesista:

```
sql Copy
SELECT p.professione, p.cod_fiscale, COUNT(*) AS numero_pazienti
FROM persone p
INNER JOIN (
    SELECT infermiere FROM ricoverati
    UNION
    SELECT anestesista FROM terapiaintensiva
) t ON t.infermiere = p.cod_fiscale OR t.anestesista = p.cod_fiscale
WHERE p.professione IN ('infermiere', 'anestesista')
GROUP BY p.cod_fiscale
ORDER BY numero_pazienti DESC
```

CLAUDE INSTANT

Query 10 - ugly schema

Livello Difficoltà: 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Ecco la traduzione in SQL della query:

```
SELECT nfrm, COUNT(*) AS num_pazienti
FROM rcvrt
WHERE nfrm IN (
    SELECT cd_fscl FROM prsn WHERE prfssn IN
    ('infermiere','anestesista')
)
GROUP BY nfrm
```

Query 10 - ugly & insert

Livello Difficoltà: 3

Esito: **Corretto**

...

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Ecco la traduzione della query:

sql

Copy

```
SELECT
    prsn.nm,
    prsn.prfssn,
    COUNT(rcvrt.d_nfzn) AS numero_pazienti
FROM prsn
LEFT JOIN rcvrt
    ON prsn.cd_fscl = rcvrt.nfrmrr
WHERE prsn.prfssn IN ('Infermiere', 'Anestesista')
GROUP BY prsn.cd_fscl, prsn.nm, prsn.prfssn;
```

Query 11

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni"

Ecco la query in SQL:

sql

Copy

```
SELECT AVG(DATEDIFF(data_guarigione, data_infezione)) AS durata_media
FROM infetti
JOIN guariti USING (id_infezione)
JOIN persone ON persona = cod_fiscale
WHERE YEAR(CURRENT_DATE) - YEAR(data_nascita) > 60
```

Query 11 - ugly schema

Livello Difficoltà: 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni"

Ecco la query SQL per calcolare la durata media del contagio prima della guarigione per le persone con più di 60 anni:

```
SELECT AVG(DATEDIFF(dt_grgn, dt_nfzn)) AS average_duration  
FROM nfft  
JOIN prsn ON prsn.cd_fscl = nfft.prsn  
WHERE DATEDIFF(CURDATE(), dt_nsct)/365 > 60
```

Query 11 - ugly & insert

Livello Difficoltà: 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni"

Questa query calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni:

sql

Copy

```
SELECT AVG(DATEDIFF(dt_dcst, dt_nfzn)) AS avg_duration
FROM nftt
JOIN prsn ON prsn.cd_fsc1 = nftt.prsn
WHERE YEAR(prsn.dt_nsct) < YEAR(CURDATE()) - 60
AND nftt.stt = 'guarito'
```

Query 12

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

Ecco la query SQL per ottenere le persone in isolamento nella regione Lombardia a dicembre 2021:

```
SELECT p.nome, p.cognome  
FROM persone p  
JOIN isolamento i ON i.id_infezione = p.id_infezione  
JOIN infetti inf ON inf.id_infezione = i.id_infezione  
JOIN comuni c ON c.codice_catastale = p.res_comune  
JOIN province prov ON prov.sigla = c.provincia  
JOIN regioni reg ON reg.nome = prov.regione  
WHERE reg.nome = 'Lombardia'  
AND MONTH(inf.data_infezione) = 12  
AND YEAR(inf.data_infezione) = 2021
```

La query effettua un JOIN tra la tabella persone, isolamento e infetti per ottenere i nomi e cognomi delle persone in isolamento. Viene quindi JOINata anche la tabella comuni per ottenere il comune di residenza, e quindi le tabelle province e regioni per filtrare solo i risultati relativi alla regione Lombardia. La clausola WHERE filtra i risultati per isolamenti avvenuti nel mese di dicembre 2021.

CLAUDE INSTANT

Query 12 - ugly schema

Livello Difficoltà: 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

Ecco la query in SQL:

```
SELECT *
FROM prsn
JOIN nfft ON prsn.cd_fscl = nfft.prsn
JOIN btzn ON prsn.ndrzz = btzn.ndrzz AND prsn.rs_cmn =
btzn.cmn
JOIN cmn ON btzn.cmn = cmn.cdc_ctstl
JOIN prvnc ON cmn.prvnc = prvnc.sgl
WHERE MONTH(nfft.dt_nfzn) = 12 AND YEAR(nfft.dt_nfzn) = 2021
AND prvnc.nm = 'Lombardia'
AND nfft.stt = 'isolamento';
```

Query 12 - ugly & insert

Livello Difficoltà: 3

Esito: **Scorretto**

...

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

```
SELECT *
FROM prsn
JOIN slmnt ON prsn.cd_fscl = slmnt.d_nfzn
JOIN nftt ON nftt.d_nfzn = slmnt.d_nfzn
JOIN cmn ON cmn.cdc_ctstl = prsn.ns_cmn
JOIN prvnc ON prvnc.sgl = cmn.prvnc
JOIN rgn ON rgn.nm = prvnc.rgn
WHERE nftt.stt = 'isolamento'
AND MONTH(nftt.dt_nfzn) = 12
AND YEAR(nftt.dt_nfzn) = 2021
AND rgn.nm = 'Lombardia'
```

Query 1

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

Per calcolare il numero di guariti in una certa data, dovremo utilizzare la tabella `guariti` che contiene le informazioni sugli individui che si sono ripresi dall'infezione. La query SQL per ottenere il numero di guariti in una specifica data sarebbe la seguente:

sql

 Copy code

```
SELECT COUNT(*) AS numero_di_guariti  
FROM guariti  
WHERE data_guarigione = 'data_desiderata';
```

chat gpt

Query 1 - ugly schema

Livello : 1

Esito: **Scorretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

La traduzione in SQL della query "Calcola il numero di guariti in una certa data" dipende dalla struttura della tabella che contiene le informazioni sui guariti e dalla modalità in cui i dati sono archiviati. Dato che non dispongo della struttura completa del database, posso solo fornire un esempio generico. Supponiamo che tu abbia una tabella chiamata "nftt" (presumo che sia una tabella contenente informazioni su persone guarite), e i campi rilevanti siano "d_nfzn" (identificativo della persona guarita) e "stt" (lo stato della guarigione, come "guarito").

Ecco come sarebbe la query SQL:

```
sql Copy code
SELECT COUNT(*) AS numero_di_guariti
FROM nftt
WHERE dt_nfzn = '2023-07-25' AND stt = 'guarito';
```

chat gpt

Query 1 - ugly & insert

Livello : 1

Esito: **Scorretto**

Traduci in sql la seguente query "Calcola il numero di guariti in una certa data".

Per calcolare il numero di guariti in una certa data, dovrà eseguire una query SQL che effettui un'operazione di conteggio sui record della tabella `nftt` (presumibilmente la tabella delle persone) in cui lo stato sia "guarito" (`stt = 'guarito'`) e la data corrisponda a quella specificata.

Supponendo che la data di interesse sia rappresentata dalla variabile `data_di_interesse`, la query SQL per ottenere il numero di guariti in quella data sarebbe la seguente:

sql

Copy code

```
SELECT COUNT(*) AS numero_di_guariti
FROM nftt
WHERE dt_nfzn = 'data_di_interesse' AND stt = 'guarito';
```

chat gpt

Query 2

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

Per calcolare il numero di farmacie che hanno aperto nel 2020, dovremo utilizzare la tabella 'farmacie' che contiene le informazioni sulle farmacie, incluso il campo 'inizio_attivita' che indica la data di apertura di ciascuna farmacia. La query SQL per ottenere il numero di farmacie aperte nel 2020 sarebbe la seguente:

sql

 Copy code

```
SELECT COUNT(*) AS numero_di_farmacie_aperte_2020
FROM farmacie
WHERE YEAR(inizio_attivita) = 2020;
```

chat gpt

Query 2 - ugly schema

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

La traduzione in SQL della query "Calcola il numero di farmacie che hanno aperto nel 2020" dipende dalla struttura della tabella che contiene le informazioni sulle farmacie e dalla modalità in cui i dati sono archiviati. Dato che non dispongo della struttura completa del database, posso solo fornire un esempio generico. Supponiamo che tu abbia una tabella chiamata "frm" (presumo che sia una tabella contenente informazioni sulle farmacie), e che il campo "nz_ttv" contenga la data di apertura delle farmacie.

Ecco come sarebbe la query SQL:

sql

Copy code

```
SELECT COUNT(*) AS numero_di_farmacie_aperte
FROM frm
WHERE YEAR(nz_ttv) = 2020;
```

chat gpt

Query 2 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il numero di farmacie che hanno aperto nel 2020".

Per calcolare il numero di farmacie che hanno aperto nel 2020, è necessario selezionare tutte le righe dalla tabella delle farmacie (``frmfc``) dove il campo `nz_ttvvt` (data di apertura) rientra nell'anno 2020 e quindi contare il numero di righe restituite. La query SQL per ottenere questa informazione sarebbe:

sql

 Copy code

```
SELECT COUNT(*) AS numero_farmacie_aperte  
FROM frmfc  
WHERE YEAR(nz_ttvvt) = 2020;
```

chat gpt

Query 3

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

Per calcolare il totale degli asintomatici nel mese di luglio del 2020, dovremo utilizzare la tabella `infetti` che contiene le informazioni sugli individui infetti, incluso il campo `stato` che indica lo stato di ciascun individuo (tra cui "asintomatico") e il campo `data_infezione` che contiene la data di infezione. La query SQL per ottenere il totale degli asintomatici nel mese di luglio del 2020 sarebbe la seguente:

sql

 Copy code

```
SELECT COUNT(*) AS totale_asintomatici
FROM infetti
WHERE stato = 'asintomatico' AND MONTH(data_infezione) = 7 AND YEAR(data_inf)
```

chat gpt

Query 3 - ugly schema

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

La traduzione in SQL della query "Calcola il totale degli asintomatici nel mese di luglio del 2020" dipende dalla struttura della tabella che contiene le informazioni sugli asintomatici e dalla modalità in cui i dati sono archiviati. Dato che non dispongo della struttura completa del database, posso solo fornire un esempio generico. Supponiamo che tu abbia una tabella chiamata "nfft" (presumo che sia una tabella contenente informazioni sulle persone con stato di salute), e che il campo "stt" contenga lo stato di salute, con valori come "asintomatico", "lieve", "grave", "critico" o "guarito", e che il campo "dt_nfzn" contenga la data in cui è stata registrata la salute.

Ecco come sarebbe la query SQL:

```
sql
Copy code

SELECT COUNT(*) AS totale_asintomatici
FROM nfft
WHERE stt = 'asintomatico' AND YEAR(dt_nfzn) = 2020 AND MONTH(dt_nfzn) = 7;
Regenerate response
```

chat gpt

Query 3 - ugly & insert

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Calcola il totale degli asintomatici nel mese di luglio del 2020".

Per calcolare il totale degli asintomatici nel mese di luglio del 2020, è necessario utilizzare la tabella `nftt` (che contiene informazioni sullo stato di salute delle persone) e filtrarla per il mese di luglio del 2020 e lo stato "asintomatico".

La query SQL per calcolare il totale degli asintomatici nel mese di luglio 2020 potrebbe essere la seguente:

sql

 Copy code

```
SELECT COUNT(*) AS totale_asintomatici
FROM nftt
WHERE YEAR(dt_nfzn) = 2020 AND MONTH(dt_nfzn) = 7 AND stt = 'asintomatico';
```

chat gpt

Query 4

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte"

Per selezionare tutte le multe fatte in un dato comune e in una specifica data, dovrà utilizzare la tabella `multe` che contiene le informazioni sulle multe emesse, inclusi i campi `comune_provenienza` e `data`. La query SQL per ottenere tutte le multe fatte in un determinato comune e in una specifica data sarebbe la seguente:

sql

 Copy code

```
SELECT *
FROM multe
WHERE comune_provenienza = 'codice_comune' AND data = 'data_desiderata';
```

chat gpt

Query 4 - ugly schema

Livello : 1

Esito: **Corretto**

Traduci in sql la seguente query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte"

La traduzione in SQL della query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte" dipende dalla struttura della tabella che contiene le informazioni sulle multe e dalla modalità in cui i dati sono archiviati. Dato che non dispongo della struttura completa del database, posso solo fornire un esempio generico. Supponiamo che tu abbia una tabella chiamata "mlt" (presumo che sia una tabella contenente informazioni sulle multe), e che i campi "cmn_prvnnz" e "dt" contengano rispettivamente il comune in cui è stata fatta la multa e la data in cui è stata emessa.

Ecco come sarebbe la query SQL:

```
sql
SELECT *
FROM mlt
WHERE cmn_prvnnz = 'NOME_COMUNE' AND dt = '2023-07-25';
```

chat gpt

Query 4 - ugly & insert

Livello : 1

Esito: **Scorretto**

Traduci in sql la seguente query "Dato un comune ed una data, seleziona tutte le multe che sono state fatte"

Per tradurre la query in SQL, assumiamo di avere una tabella chiamata "multe" con le seguenti colonne: "comune" per il comune in cui è stata fatta la multa e "data" per la data in cui è stata fatta la multa. La tabella potrebbe avere anche altre colonne di cui non teniamo conto per questa query.

La query tradotta in SQL sarebbe:

sql

 Copy code

```
SELECT *
FROM multe
WHERE comune = 'nome_comune' AND data = 'data_selezionata';
```

chat gpt

Query 5

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per ogni comune".

Per ottenere il numero di abitanti per ogni comune dalla tabella `persone`, puoi utilizzare la seguente query SQL:

sql



Copy code

```
SELECT comuni.nome AS comune, COUNT(*) AS numero_abitanti
FROM comuni
LEFT JOIN persone ON comuni.codice_catastale = persone.nas_comune
GROUP BY comuni.nome;
```

chat gpt

Query 5 - ugly schema

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per ogni comune".

La seguente query SQL seleziona il numero di abitanti per ogni comune:

sql

 Copy code

```
SELECT cmm.cdc_ctstl AS CodiceComune, cmm.nm AS NomeComune, COUNT(prsn.cd_fs  
FROM cmm  
LEFT JOIN prsn ON cmm.cdc_ctstl = prsn.ns_cmn  
GROUP BY cmm.cdc_ctstl, cmm.nm;
```

chat gpt

Query 5 - ugly schema & insert

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Selezione il numero di abitanti per ogni comune".

Per ottenere il numero di abitanti per ogni comune, è necessario utilizzare la funzione di aggregazione `COUNT()` insieme a una clausola `GROUP BY`. La query SQL sarebbe la seguente:

sql

Copy code

```
SELECT cmn.cdc_ctstl AS CodiceComune, cmn.nm AS NomeComune, COUNT(prsn.cd_fs
FROM cmn
LEFT JOIN prsn ON cmn.cdc_ctstl = prsn.ns_cmn
GROUP BY cmn.cdc_ctstl, cmn.nm;
```

chat gpt

Query 5.5

Livello : 2

Esito: **Corretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per comune di residenza".

Per ottenere il numero di abitanti per ogni comune di residenza dalla tabella `persone`, puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT comuni_residenza.nome AS comune_residenza, COUNT(*) AS numero_abitant
FROM comuni AS comuni_residenza
JOIN persone ON comuni_residenza.codice_catastale = persone.res_comune
GROUP BY comuni_residenza.nome;
```

chat gpt

Query 5.5 - ugly schema

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per comune di residenza".

Per selezionare il numero di abitanti per comune di residenza, dobbiamo presumere che i dati sul numero di abitanti siano memorizzati nella tabella `cmn` e che ci sia un collegamento tra la tabella `cmn` e la tabella `prsn` per stabilire la residenza di ciascuna persona. Poiché nella tabella `prsn` c'è una colonna `ns_cmn` che fa riferimento al comune di residenza, possiamo usarla per ottenere il numero di abitanti per ogni comune di residenza.

La query SQL sarebbe la seguente:

```
sql
SELECT cmn.nm AS Comune, COUNT(*) AS Numero_Abitanti
FROM cmn
JOIN prsn ON cmn.cdc_ctstl = prsn.ns_cmn
GROUP BY cmn.nm;
```

chat gpt

Query 5.5 - ugly schema & insert

Livello : 2

Esito: **Scorretto**

Traduci in sql la seguente query "Seleziona il numero di abitanti per comune di residenza".

La query SQL per selezionare il numero di abitanti per comune di residenza si basa sulla tabella `nftt`, assumendo che questa tabella contenga le informazioni sulla popolazione di ciascun comune. La query sarà la seguente:

sql

Copy code

```
SELECT prsn AS ComuneDiResidenza, COUNT(d_nfzn) AS NumeroAbitanti  
FROM nftt  
GROUP BY prsn;
```

chat gpt

Query 6

Livello : 2

Esito: **Corretto***

evita il join, manca il nome

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

Per calcolare la quantità venduta e il ricavo per ogni prodotto dalla tabella `vendite` e ordinarli per quantità venduta in ordine decrescente, puoi utilizzare la seguente query SQL:

sql

 Copy code

```
SELECT prodotto, SUM(quantita) AS quantita_venduta, SUM(totale) AS ricavo
FROM vendite
GROUP BY prodotto
ORDER BY quantita_venduta DESC;
```

chat gpt

Query 6 - ugly schema

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

Per calcolare la quantità venduta e il ricavo per ogni prodotto e ordinare i risultati per quantità venduta in ordine decrescente, dobbiamo presumere che i dati relativi alle vendite siano memorizzati nella tabella `vndt`, e i dettagli dei prodotti siano memorizzati nella tabella `prdtt`.

La query SQL sarebbe la seguente:

```
sql Copy code
SELECT prdtt.nm AS Prodotto, SUM(vndt.qntt) AS Quantita_Venduta, SUM(vndt.tt
FROM prdtt
JOIN vndt ON prdtt.cdc = vndt.prdtt
GROUP BY prdtt.nm
ORDER BY Quantita_Venduta DESC;
```

chat gpt

Query 6 - ugly schema & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Per ogni prodotto, calcola in che quantità e' stato venduto ed il ricavo. Ordina quindi per quantità decrescente"

Per tradurre la query in SQL, assumeremo l'esistenza di due tabelle:

.. Tabella "prodotti" con le seguenti colonne:

.. id_prodotto

.. nome_prodotto

.. Tabella "vendite" con le seguenti colonne:

.. id_prodotto

.. quantità_venduta

.. ricavo

La query tradotta in SQL sarà la seguente:

```
sql Copy code
SELECT p.nome_prodotto AS prodotto,
       v.quantità_venduta AS quantità,
       v.ricavo
  FROM prodotti p
 JOIN (
    SELECT id_prodotto,
           SUM(quantità_venduta) AS quantità_venduta,
           SUM(ricavo) AS ricavo
      FROM vendite
     GROUP BY id_prodotto
   ) v ON p.id_prodotto = v.id_prodotto
 ORDER BY v.quantità_venduta DESC;
```

chat gpt

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Per tradurre la query in SQL, dovrà utilizzare le tabelle fornite nel dump del database, in particolare la tabella 'infetti'. Assicurati di avere già creato il database e le tabelle descritte nel dump.

La query SQL per mostrare il numero di infermieri che si sono infettati sarà simile a questa:

```
sql Copy code
SELECT COUNT(*) AS numero_infermieri_infettati
FROM infetti
WHERE stato = 'asintomatico' OR stato = 'lieve' OR stato = 'grave' OR stato
```

Query 7

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Per mostrare il numero di infermieri che si sono infettati dalla tabella 'ricoverati', puoi utilizzare la seguente query SQL:

```
sql
```

Copy code

```
SELECT COUNT(DISTINCT infermiere) AS numero_infermieri_infettati
FROM ricoverati
WHERE id_infezione IS NOT NULL;
```

chat gpt

Query 7 - ugly schema

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Per mostrare il numero di infermieri che si sono infettati, dobbiamo presumere che i dati relativi agli infermieri e alle infezioni siano memorizzati nella tabella `trpntnsv`. Inoltre, dovremo assumere che ci sia una colonna nella tabella `trpntnsv` che indica se una persona è un infermiere e un'altra colonna che indica lo stato dell'infezione.

La query SQL sarebbe la seguente:

sql

Copy code

```
SELECT COUNT(*) AS Numero_Infermieri_Infettati  
FROM trpntnsv  
WHERE rsprtr = 'Infermiere' AND nstsst = 'Infettato';
```

chat gpt

Query 7 - ugly schema & insert

Livello : 2

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra il numero di infermieri che si sono infettati"

Per mostrare il numero di infermieri che si sono infettati, dovresti utilizzare la tabella `nftt` che contiene le informazioni sugli infermieri e la colonna `stt` per indicare lo stato di infezione. Nella tua struttura del database, sembra che la colonna `stt` sia presente nella tabella `nftt`.

La query SQL per ottenere il numero di infermieri infettati sarebbe la seguente:

```
sql
SELECT COUNT(*) AS NumeroDiInfermieriInfettati
FROM nftt
WHERE stt = 'infetto';
```

chat gpt

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Per contare il numero di ospedali per provincia dalla tabella `ospedali` e ottenere la lista ordinata in base al numero di ospedali, puoi utilizzare la seguente query SQL:

```
sql
Copy code
SELECT comune, COUNT(*) AS numero_ospedali
FROM ospedali
GROUP BY provincia
ORDER BY numero_ospedali;
```

Query 8

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Per tradurre la query in SQL, dovrà utilizzare le tabelle fornite nel dump del database, in particolare la tabella `ospedali` e `comuni`. Assicurati di avere già creato il database e le tabelle descritte nel dump.

La query SQL per contare il numero di ospedali per provincia e restituire la lista ordinata sarà simile a questa:

```
sql
Copy code
SELECT c.provincia, COUNT(o.partita_iva) AS numero_ospedali
FROM comuni c
LEFT JOIN ospedali o ON c.codice_catastale = o.comune
GROUP BY c.provincia
ORDER BY numero_ospedali DESC;
```

Regenerate response

chat gpt

Query 8 - ugly schema

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Per contare il numero di ospedali per provincia e restituire la lista ordinata, dobbiamo presumere che i dati relativi agli ospedali siano memorizzati nella tabella `spdl`, e che ci sia una colonna nella tabella `spdl` che indica la provincia di appartenenza di ciascun ospedale.

La query SQL sarebbe la seguente:

```
sql Copy code
SELECT cmn.prvnc AS Provincia, COUNT(*) AS Numero_Ospedali
FROM spdl
JOIN cmn ON spdl.cmn = cmn.cdc_ctstl
GROUP BY cmn.prvnc
ORDER BY Numero_Ospedali DESC;
```

chat gpt

Query 8 - ugly schema & insert

Livello : 2

Esito: **Corretto**

Traduci in SQL la seguente query: "Conta il numero di ospedali per provincia e ritorna la lista ordinata"

Per contare il numero di ospedali per provincia e ottenere la lista ordinata, puoi utilizzare la seguente query SQL:

sql

Copy code

```
SELECT prvnc.nm AS provincia, COUNT(spdl.cdc) AS numero_ospedali
FROM prvnc
LEFT JOIN cmn ON prvnc.sgl = cmn.prvnc
LEFT JOIN spdl ON cmn.cdc_ctstl = spdl.cmn
GROUP BY prvnc.nm
ORDER BY numero_ospedali DESC;
```

chat gpt

Query 9

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

Per calcolare per ogni regione il numero di guariti, decessi, attualmente positivi e casi totali, dovremo utilizzare le operazioni di aggregazione insieme a una clausola GROUP BY per raggruppare i dati per regione. Dato che non abbiamo accesso ai dati completi e alle tabelle specifiche del database, fornirò una query generica basata sulla struttura delle tabelle presenti nel dump SQL fornito in precedenza. Tieni presente che potresti dover adattare questa query al tuo database specifico se le tabelle o i campi hanno nomi diversi.

Assumendo che le tabelle coinvolte siano 'regioni', 'guariti', 'deceduti', e 'infetti', la query per calcolare i dati richiesti sarebbe la seguente:

```
sql Copy code
SELECT
    r.nome AS regione,
    COUNT(g.id_infezione) AS guariti,
    COUNT(d.id_infezione) AS decessi,
    COUNT(i.id_infezione) AS attualmente_positivi,
    COUNT(i.id_infezione) + COUNT(g.id_infezione) + COUNT(d.id_infezione) AS casi_totali
FROM regioni r
LEFT JOIN guariti g ON r.nome = g.nome_regione
LEFT JOIN deceduti d ON r.nome = d.nome_regione
LEFT JOIN infetti i ON r.nome = i.nome_regione
GROUP BY r.nome;
```

chat gpt

Query 9 - ugly schema

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

cmn (comune) ha provincia e non rgn (regione)

```
sql
Copy code

SELECT
    r.`nm` AS Regione,
    COUNT(CASE WHEN n.`stt` = 'guarito' THEN 1 END) AS Guariti,
    COUNT(CASE WHEN n.`stt` = 'decesso' THEN 1 END) AS Decessi,
    COUNT(CASE WHEN n.`stt` = 'attualmente positivo' THEN 1 END) AS AttualmentePositivo,
    COUNT(*) AS CasiTotali
FROM
    `rgn` r
LEFT JOIN
    `cmn` c ON r.`nm` = c.`rgn`
LEFT JOIN
    `btzn` b ON c.`cdc_ctstl` = b.`cmn`
LEFT JOIN
    `nftt` n ON b.`ndrzz` = n.`d_nfzn`
GROUP BY
    r.`nm`
ORDER BY
    r.`nm`;
```

chat gpt

Query 9 - ugly schema & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali."

Per calcolare per ogni regione il numero di guariti, di decessi, di attualmente positivi e i casi totali, assumiamo che tu abbia due tabelle: una tabella chiamata "Regioni" con le informazioni delle regioni e un'altra tabella chiamata "Dati_Covid" con le informazioni sui singoli casi di COVID-19, collegata alla tabella delle regioni tramite una chiave esterna "id_regione".

La query SQL sarebbe la seguente:

```
sql Copy code
SELECT r.nome_region AS Regione,
       SUM(dc.guariti) AS Guariti,
       SUM(dc.decessi) AS Decessi,
       SUM(dc.attualmente_positivi) AS AttualmentePositivi,
       COUNT(dc.id_caso) AS CasiTotali
  FROM Regioni r
 LEFT JOIN Dati_Covid dc ON r.id_regione = dc.id_regione
 GROUP BY r.nome_region;
```

Query 10

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Per tradurre la query in SQL, dovrà utilizzare una combinazione di JOIN e GROUP BY per ottenere il numero di pazienti ricoverati affidati a ciascun infermiere o anestesista.

Presumendo che l'informazione sui pazienti ricoverati sia memorizzata nella tabella 'ricoverati', e i dati sugli infermieri e anestesi si siano presenti nelle tabelle 'persone' e 'terapiaintensiva', ecco la query:

```
sql Copy code
SELECT p.cod_fiscale, p.nome, p.cognome, COUNT(r.id_infezione) AS numero_paz
FROM persone p
LEFT JOIN terapiaintensiva t ON p.cod_fiscale = t.infermiere OR p.cod_fiscale
LEFT JOIN ricoverati r ON t.id_infezione = r.id_infezione
GROUP BY p.cod_fiscale, p.nome, p.cognome;
```

Query 10 - ugly schema

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Per tradurre la query in SQL, dovremo utilizzare le tabelle fornite nel dump del database e identificare come associare gli infermieri o anestesiisti con i pazienti ricoverati. Dato che la tabella `prsn` contiene informazioni sugli infermieri o anestesiisti e la tabella `rcvrt` contiene informazioni sui pazienti ricoverati e l'affidamento degli stessi, utilizzeremo un JOIN per ottenere i risultati desiderati. Di seguito è riportata la traduzione della query in SQL:

```
sql Copy code
SELECT
    p.`cgnm` AS Cognome,
    p.`nm` AS Nome,
    COUNT(r.`d_nfzn`) AS NumeroPazientiAffidati
FROM
    `prsn` p
LEFT JOIN
    `rcvrt` r ON p.`cd_fscl` = r.`nfrmrr`
GROUP BY
    p.`cd_fscl`, p.`cgnm`, p.`nm`
HAVING
    p.`prfssn` IN ('infermiere', 'anestesista')p. `prfssn` IN ('infermiere', 'anestesista')
```

inserisce in having clausola non selettiva.. ma se l'avesse messa? manca comunque la tabella di terapia intensiva (questa' e' quasi una 3.5)

Query 10 - ugly schema & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Per ogni infermiere o anestesista, seleziona il numero di pazienti ricoverati che gli sono stati affidati."

Per tradurre la query in SQL, supponendo che ci siano due tabelle coinvolte: una tabella chiamata "infermieri" con i dati degli infermieri e una tabella chiamata "pazienti" con i dati dei pazienti e il campo "id_infermiere" per indicare quale infermiere si occupa di ciascun paziente, puoi utilizzare la seguente query:

```
sql Copy code
SELECT
    id_infermiere,
    COUNT(*) AS numero_pazienti_affidati
FROM
    pazienti
WHERE
    id_infermiere IS NOT NULL
GROUP BY
    id_infermiere
ORDER BY
    id_infermiere;
```

Query 11

Livello : 3

Esito: **Corretto**

Traduci in SQL la seguente query: "Calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni"

Per calcolare la durata media del contagio prima della guarigione per le persone con più di 60 anni, dovrà fare uso della tabella `infetti` che contiene le informazioni sulle persone infette e la tabella `guariti` che contiene le informazioni sui guariti. Assumendo che la data di infezione (`data_infezione`) sia presente nella tabella `infetti` e la data di guarigione (`data_guarigione`) sia presente nella tabella `guariti` , ecco la query SQL:

sql

 Copy code

```
SELECT AVG(DATEDIFF(g.data_guarigione, i.data_infezione)) AS durata_media_cc
FROM infetti i
JOIN guariti g ON i.id_infezione = g.id_infezione
JOIN persone p ON i.persona = p.cod_fiscale
WHERE TIMESTAMPDIFF(YEAR, p.data_nascita, CURDATE()) > 60;
```

Query 11 - ugly schema

Livello : 3

Esito: **Scorretto**

Per calcolare la durata media del contagio prima della guarigione per le persone con più 60 anni, dobbiamo utilizzare le tabelle pertinenti e calcolare la differenza tra la data di guarigione (`dt_nfzn`) e la data di contagio (`dt_dcss`) per le persone con più di 60 anni.

Di seguito è riportata la traduzione della query in SQL:

```
sql Copy code
SELECT
    AVG(DATEDIFF(n.`dt_nfzn`, d.`dt_dcss`)) AS DurataMediaContagio
FROM
    `nftt` n
INNER JOIN
    `dcdt` d ON n.`d_nfzn` = d.`d_nfzn`
INNER JOIN
    `prsn` p ON n.`prsn` = p.`cd_fscl`
WHERE
    p.`dt_nsct` <= DATE_SUB(CURDATE(), INTERVAL 60 YEAR)
    AND n.`stt` = 'guarito';
```

Query 11 - ugly schema & insert

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Calcola la durata media del contagio prima della guarigione per le persone con più di 60 anni"

Per calcolare la durata media del contagio prima della guarigione per le persone con più di 60 anni, assumiamo di avere un database con due tabelle: una tabella chiamata "Persone" con le informazioni anagrafiche delle persone e un'altra tabella chiamata "Contagi" con le informazioni sui casi di contagio e guarigione. La tabella "Contagi" dovrebbe contenere almeno le seguenti colonne: "persona_id" (identificativo unico della persona), "data_contagio" (data di inizio del contagio), "data_guarigione" (data di guarigione).

La query SQL per calcolare la durata media del contagio prima della guarigione per le persone con più di 60 anni è la seguente:

```
sql Copy code
SELECT AVG(DATEDIFF(data_guarigione, data_contagio)) AS durata_media_contagi
FROM Persone p
JOIN Contagi c ON p.id = c.persona_id
WHERE p.età > 60;
```

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

Per tradurre la query in SQL, dovremo utilizzare la tabella delle persone ('**persone**') e la tabella di isolamento ('**isolamento**') insieme alla tabella delle regioni ('**regioni**') per filtrare le persone che erano in isolamento nella regione Lombardia a dicembre 2021.

Supponendo che la data di inizio dell'isolamento sia rappresentata da una colonna 'data_inizio_isolamento' nella tabella '**isolamento**', possiamo scrivere la query come segue:

```
sql Copy code
SELECT p.*
FROM persone p
INNER JOIN isolamento i ON p.cod_fiscale = i.id_infezione
INNER JOIN regioni r ON p.res_comune = r.nome
WHERE i.data_inizio_isolamento >= '2021-12-01'
AND i.data_inizio_isolamento <= '2021-12-31'
AND r.nome = 'Lombardia';
```

Query 12

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

Per ottenere le persone che erano in isolamento nella regione Lombardia a dicembre 2021, dovrà combinare le informazioni dalle tabelle '**isolamento**', '**infetti**', e '**persone**'.

Assumendo che la data di infezione ('**data_infezione**') sia presente nella tabella '**infetti**', e la data di isolamento sia presente nella tabella '**isolamento**', e che la regione di residenza ('**res_comune**') sia presente nella tabella '**persone**', ecco la query SQL:

```
sql Copy code
SELECT p.*
FROM persone p
JOIN infetti i ON p.cod_fiscale = i.persona
JOIN isolamento iso ON i.id_infezione = iso.id_infezione
WHERE iso.data = '2021-12-31' AND p.res_comune = 'lomb';
```

chat gpt

Query 12

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

chat gpt

```
sql Copy code
SELECT
    p.`cgnm` AS Cognome,
    p.`nm` AS Nome,
    p.`dt_nsct` AS DataNascita,
    n.`dt_nfzn` AS DataIsolamento
FROM
    `prsn` p
INNER JOIN
    `nftt` n ON p.`cd_fscl` = n.`prsn`
INNER JOIN
    `dcdt` d ON n.`d_nfzn` = d.`d_nfzn`
INNER JOIN
    `cnn` c ON d.`dt_dcst` >= '2021-12-01' AND d.`dt_dcst` <= '2021-12-31' AND c.`cnn` = d.`cnn`
INNER JOIN
    `rgn` r ON c.`rgn` = r.`nm`
WHERE
    n.`stt` = 'ttualmente positivo' AND r.`nm` = 'Lombardia';
```

Query 12

Livello : 3

Esito: **Scorretto**

Traduci in SQL la seguente query: "Mostra le persone che a dicembre 2021 erano in isolamento nella regione Lombardia"

Per tradurre la query in SQL, assumiamo che tu abbia una tabella chiamata "persone" con i seguenti campi: "nome", "cognome", "data_di_nascita", "regione", "status_isolamento".

La query SQL per mostrare le persone che a dicembre 2021 erano in isolamento nella regione Lombardia sarebbe la seguente:

```
sql
SELECT nome, cognome
FROM persone
WHERE regione = 'Lombardia' AND status_isolamento = 'in isolamento' AND data
```

In questa query: