

Domande SQL con Risposte Corrette e Motivazioni

Domande con Risposte Corrette e Motivazioni

Domanda 1: Il predicato di join per un rapporto fra tabelle che simuli l'ereditarietà

La risposta corretta è la prima: **si ottiene eguagliando le chiavi primarie delle tabelle coinvolte**. Questo perché, come illustrato nel testo, il predicato di INNER JOIN per un rapporto 1-1 che simula l'ereditarietà si ottiene "semplicemente eguagliando le chiavi primarie", stabilendo la condizione `Person.ID = Teacher.ID`. Le due tabelle condividono lo stesso identificativo poiché rappresentano la stessa entità.

Domanda 2: La forma esplicita nell'inner join

La risposta corretta è la quarta: **è specificata nel FROM**. La motivazione risiede nel fatto che la forma esplicita dell'INNER JOIN usa la sintassi `SELECT * FROM T1 INNER JOIN T2 ON (PREDICATO DI JOIN)`, dove la condizione di join con la clausola ON fa parte integrante della clausola FROM, separata dalla SELECT e dal WHERE. Questa separazione permette di distinguere i predicati di join dalle condizioni di dominio.

Domanda 3: Vogliamo trovare nome e cognome di tutte le persone e degli insegnanti vogliamo il salario, il tutto con una sola query

La risposta corretta è la quarta: **un LEFT JOIN**. Il motivo è che desiderando visualizzare tutte le persone, incluse quelle che non sono insegnanti, è necessario utilizzare il LEFT JOIN. Questo costrutto preserva tutte le righe della tabella Person e aggiunge i dati di Teacher quando presenti, mostrando NULL per il salario di chi non risulta essere un insegnante. Un INNER JOIN escluderebbe invece tutte le persone prive di un corrispettivo nella tabella Teacher.

Domanda 4: Cosa si intende con condizione di dominio?

La risposta corretta è la terza: **una forma di selezione che esclude i predicati di join**. La spiegazione è che, come evidenziato nel testo, l'uso della forma esplicita con INNER JOIN "permette di separare adeguatamente i predicati di join, che rappresentano condizioni strutturali, dalle condizioni di ricerca dei dati, che sono appunto le condizioni di dominio". Le condizioni di dominio sono quindi le clausole WHERE utilizzate per filtrare i dati in base a criteri specifici, distinguendosi nettamente dai predicati di join che definiscono esclusivamente le relazioni strutturali tra le tabelle.

Domanda 5: Col left join

La risposta corretta è la seconda: **le righe della tabella a sinistra non verranno scartate dal LEFT JOIN**. Questo accade perché il LEFT JOIN, per sua definizione, preserva tutte le righe della tabella posta a sinistra dell'operatore, indipendentemente dal fatto che queste partecipino o meno alla relazione con la tabella di destra. Il comportamento può essere ricordato come "preserva tutte le righe a sinistra", ma è importante precisare che le condizioni di dominio poste nella clausola WHERE possono comunque scartare righe anche dopo l'esecuzione del LEFT JOIN.

Quiz Fine Modulo 7: Domande Aggiuntive

Domanda 1: Il left join

La risposta corretta è la seconda: **è utile per i rapporti di ereditarietà e per i rapporti in cui vogliamo conservare anche i padri senza figli**. La motivazione risiede nel fatto che il LEFT JOIN preserva tutte le righe della tabella padre (a sinistra), risultando particolarmente utile nei rapporti di ereditarietà dove non tutte le entità della superclasse appartengono necessariamente alla sottoclasse. Nell'esempio del materiale, il LEFT JOIN permette di visualizzare anche il signor B, che non è un insegnante, preservando la riga di Person anche in assenza di un corrispettivo in Teacher.

Domanda 2: Il left join (combinabilità)

La risposta corretta è la terza: **è combinabile con inner join e condizioni di dominio**. La dimostrazione si trova nell'esempio del testo dove viene utilizzata una query che combina INNER JOIN tra Person e City con LEFT JOIN verso Teacher, il tutto filtrato da una condizione di dominio `WHERE CITY.NAME = 'MILANO'`. Il LEFT JOIN può quindi lavorare insieme ad altri tipi di join e a condizioni di filtraggio nella clausola WHERE.

Domanda 3: Simulando la relazione di ereditarietà, il predicato di join si scrive tipicamente come

La risposta corretta è la prima: **chiave primaria del padre = chiave primaria del figlio**. Nel caso specifico dell'ereditarietà simulata tra Person e Teacher, il predicato di join è `Person.ID = Teacher.ID`. Entrambe le tabelle utilizzano la stessa colonna ID come chiave primaria, poiché la tabella Teacher non ha un identificativo autonomo ma eredita quello dalla tabella Person attraverso la chiave esterna che coincide con la chiave primaria.

Domanda 4: SQL e l'ereditarietà

La risposta corretta è la terza: **permette di simularla con le normali relazioni fra tabelle**. Il testo chiarisce esplicitamente che "quello che vogliamo fare invece è simulare l'ereditarietà (che non esiste nei DBMS SQL standard) con un rapporto 1-1 opzionale". I database SQL standard non dispongono di un costrutto nativo per l'ereditarietà, pertanto questa viene simulata attraverso le normali relazioni tra tabelle utilizzando chiavi esterne e vincoli di integrità referenziale.