Risposte Teoria Elementi di Basi di Dati

Nell'ambito del controllo dell'accesso in SQL, indicare il significato e l'utilizzo dei *references*.

Il costrutto *references* nel controllo dell' accesso in SQL, permette che venga fatto riferimento alla risorsa nell' ambito della definizione dello schema di una tabella. La sua utilità si manifesta nel momento in cui le politiche di reazione di un vincolo di integrità referenziale vengono messe in atto con la possibilità di un' effetto sulle tabelle esterne.

Con riferimento al modello ER, spiegare il concetto di entità debole e fornirne un esempio.

L' entità debole è quell' entità la qui chiave contiene attributi che non appartengono all' entità stessa ma ad un' altra a cui è legata tramite una relazione.
Esempio:
AZIENDA(NomeAzienda, NrDipendenti)
DIPENDENTE(CodiceDip, NomeAzienda, Nome, Cognome, Mansione)
NomeAzienda compare nella chiave di DIPENDENTE la quale quindi è un' entità debole.

Con riferimento a SQL, spiegare il significato delle funzioni *coalesce* e *nullif* e fornire un esempio per ognuna.

Le funzioni *coalesce* e *nullif* vengono utilizzate in SQL come funzioni condizionali per eseguire delle selezione su campi che possono assumere il valore NULL.
In particolare la funzione *coalesce* restituisce il valore specificato negli argomenti ogni volta che incontra il valore NULL; la funzione *nullif* invece, restituisce il valore NULL quando incontra il valore specificato altrimenti restituisce il valore dell' espressione.
Esempio:

```
select Matr,Cognome,Nome,coalesce(CdL,'non ins.')
from Studenti
```

La funziona restituirà il valore "non ins." ogni volta che incontra il valore NULL.

```
select Matr,Cod-corso,Data,nullif(Voto,'ritirato')
from Esami
```

Utilizza il valore NULL in caso il campo Voto della tupla valga "ritirato".

Nell'ambito del linguaggio di interrogazione SQL, dire se esistono viste non modificabili, spiegando il perchè fornendo, se esiste, un esempio.

No, non esistono viste non modificabili, anche se sono state progettate principalmente per essere interrogate. C'è da dire però, che le modifiche sulle viste vengono sottoposte a diverse restrizioni, come ad esempio il fatto che una modifica su una vista deve essere applicabile univocamente alle tabelle sottostanti.

Dire cosa si intende per componente intensionale e componente estensionale di una base di dati.

Lo schema di una Base di Dati è considerata la componente intensionale e costituisce la struttura che descrive i dati e le loro relazioni; l'istanza di una Base di Dati è la componente estensionale e costituisce ilo valore dei dati contenuti nella Base di Dati in un determinato momento.

Nell'ambito del modello relazionale, elencare e descrivere le politiche di relazione che possono essere associate al vincolo di integrità referenziale in SQL.

- Vincolo INTRARELAZIONALE - soddisfatto da singole relazioni
- Vincolo di Tupla - valutato su ciascuna tupla indipendentemente dalle altre
- Vincolo di Dominio - deve rientrare in un range di valori
- Vincolo Interrelazionale - se coinvolge più relazioni

Nell'ambito del modello E-R, definire il concetto di identificatore esterno spiegando quando questo è necessario.

L'identificatore esterno è utilizzato quando si vuole identificare una certa entità attraverso un'altra entità esterna: ad esempio se vogliamo identificare uno studente tramite l'attributo Matricola e l'entità UNIVERSITA dobbiamo utilizzare un identificatore esterno; in questo modo quello studente potrà anche avere la stessa matricola registrata in più università.

Dire se la rappresentazione ER è sufficiente per rappresentare tutti i vincoli di una Base di Dati. In caso di risposta negativa, illustrare un esempio di vincolo non rappresentabile.

Lo schema ER non è sufficente perchè alcuni vincoli non possono essere espressi tramite lo schema: solitamente si ricorre ad una descrizione letterale allegata come documentazione aggiuntiva allo schema. Un esempio potrebbe essere l'attributo stipendio sull'entità OPERAIO che deve essere minore di quello dell'entità CAPO REPARTO: inesprimibile con un grafico ER.

Spiegare il concetto di integrità referenziale e fornire un esempio.

Per Integrità referenziale si intende che X attributi di una relazione R1 siano chiave primaria nella relazione R2, ovvero le due relazioni siano collegate tramite questi attributi in maniera univoca.
Un esempio potrebbe essere “Anagrafica Cittadino” con “Città italiane” collegate ad esempio tramite il campo CAP della città .

Definire i concetti di chiave, superchiave e chiave primaria (formalizzazione e spiegazione concisa).

La chiave è un attributo di una entità (un campo sul database) che identifica una determinata entità in base al suo valore. Se l'entità viene identificata solo tramite un campo si parla di chiave primaria e il valore di quel campo è assoluto ed unico solo per quella entità: nessun altro avrà lo stesso valore su quel campo.
Una superchiave è un insieme di K attributi che identificano quell'oggetto e come insieme di valori sono unici per identificare univocamente quellì'oggetto. Potremmo anche parlare di Superchiave Minimale se la superchiave non contiene sottoinsieme di chiavi primarie, ovvero se gli attributi della superchiave non rappresentano già una chiave primaria.

Cosa è l'occorrenza di una entità?

È un oggetto della classe che l'Entità rappresenta, ad esempio Milano, Roma, Palermo sono occorrenze della entità Città; non bisogna fare l'errore di considerarlo come un valore che identifica l'oggetto, ma è l'oggetto stesso.
L'occorrenza Milano avrà un campo ID='Milano' che identifica l'istanza dell'occorrenza.

Illustrare l’architettura ANSI/SPARC indicando il significato dei diversi livelli.

Architettura standardizzata (ANSI/SPARC) per DBMS fa riferimento a tre livelli (ognuno con un proprio schema):

- **Schema esterno**: descrizione, per mezzo del modello logico, di una porzione della base di dati di interesse (riflette la vista sui dati di un particolare utente o gruppo di utenti)
- **Schema logico**: descrizione dell’intera base di dati per mezzo del modello logico
- **Schema interno**: rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione

Nell’ambito del modello Entità-Relazione, definire il concetto di entità debole.

Un’entità debole è una entità la cui chiave include riferimenti ad altre entità tramite relazioni.
Esempio:
la voce di fattura è identificata dalla fattura che la contiene e dalla posizione della voce stessa nella fattura.

Illustrare e commentare brevemente le diverse fasi del ciclo di vita di una base di dati, specificando cosa ogni singola fase prende in ingresso e cosa produce. Nella descrizione, discutere anche i diversi passi della progettazione.

Durante la prgettazione di una base di dati è importante tener conto del ciclo di vita di un sistema informativo, composto da:

1. **Studio di fattibilità**: si prendono in considerazione tutte le possibili implementazioni realizzabili, si calcolano i relativi costi implementativi e si danno priorità alle varie fasi di sviluppo;
2. **Raccolta e analisi dei dati**: si prendono in considerazione tutti i dati che dovranno essere gestiti e come e quali utenti dovranno accedere agli stessi; in questa fase si produce anche uno schema astratto della base di dati. Vengono inoltre stabiliti i requisiti HW e SW del sistema informativo.
3. **Progettazione**: in prima analisi dei dati, ovvero si progetta la componente intensionale della base di dati; in seconda analisi si progettano i programmi applicativi che andranno ad interagire con i dati.
4. **Implementazione**: viene creato il database su un server di Database e viene scritto il codice dei programmi applicativi.
5. **Test e validazione**: viene testato il funzionamento del sistema informativo ed in caso di bug vengono subito cercate delle soluzioni.

6. **Funzionamento**: il sistema informativo viene utilizzato per il suo scopo originale ed in caso di corretto funzionamento necessita solo di azioni di gestione e mantenimento.

Spiegare i concetti di Indipendenza dei dati fisica e logica.

- **Indipendenza fisica:** il livello esterno e logico sono indipendenti dal livello fisico del database; modifiche fisiche di allocazione e memorizzazione non influiscono sulle applicazioni che fanno riferimento al modello logico (grazie al DBMS). Modifiche alle strutture dei dati non influiscono gli utenti del database.
- **Indipendenza Logica:** permetter di aggiungere o modificare gli schemi esterni senza modificare lo schema logico; permette inoltre di modificare lo schema logico mantenendo inalterati gli schemi esterni visibili agli utenti.

Descrivere le varie Strategie di progetto (Concettuale)

Lo sviluppo di uno schema concettuale :

- **Strategia Top-Down**: lo schema concettuale viene prodotto raffinando nel corso del tempo uno schema iniziale con pochi concetti e molto astratti; le trasformazioni da una fase all'altra avvengono tramite le cosiddette primitive di trasformazione top-down; casi in cui potrebbe avvenire è quando da un concetto generale se ne formano 2 specifici.
  VANTAGGIO: permette di evitare la descrizione dei dettagli nella fase iniziale.
- **Strategia Botton-Up**: le specifiche iniziali vengono via via suddivise in componenti sempre più frammentati e piccoli, sino ad arrivare a singoli concetti.
  VANTAGGIO: si adatta ad una decomposizione del problema in problemi più semplici.
- **Strategia Inside-Out**: è una particolare estensione del metodo botton-up, perchè partendo da solo alcuni concetti iniziali ci si estende a macchia d'olio su tutti i problemi che riguardano il progetto tenendosi comunque in relazione con i concetti iniziali.
  VANTAGGIO: permette di decomporre il problema esaminando di volta in volta le specifiche progettuali.
- **Strategia Mista**: è una strategia di incrocio fra la Botton-Up e la Top-Down (e in parte anche della Inside-Out).
  VANTAGGIO: è molto flessibile perchè permette di astrarre i componenti principali ma allo stesso tempo di averne uno scheletro descrittivo.

Descrivere i punti per giudicare la qualità di uno schema concettuale.

Vanno garantite alcune proprietà e le più importanti sono:

- **Correttezza**: quando usa in modo corretto i costrutti messi a disposizione dal modello concettuale utilizzato per la progettazione concettuale (ad esempio del modello E-R).
- **Completezza**: quando descrive tutti i dati e tutte le azioni che devono essere eseguite.
- **Leggibilità**: quando rappresenta i requisiti in maniera naturale e facilmente comprensibile; lo schema dovrebbe essere autoesplicativo.
- **Minimalità**: quando tutte le specifiche sono rappresentate una sola volta nello schema (non vi è quindi nessuna ripetizione inutile di informazione – assenza di ridondanza).

Descrivere una metodologia generale di Progettazione Concettuale.

1. **Analisi dei requisiti:**
   - Costruzione di un glossario dei termini
   - Analisi dei requisiti ed eliminazione delle ambiguità
   - Raggruppamento dei requisiti in gruppi omogenei
2. **Passo base:**
   - Individuazione dei concetti più rilevanti e rappresentazione del loro scheletro
3. **Passo di decomposizione:**
   - Decomposizione dei requisiti con riferimento allo scheletro
4. **Passo iterativo:**
   - Raffina i concetti ed aggiunge nuovi concetti per tutti i sotto-schemi
5. **Passo di integrazione:**
   - Integrazione di tutti i sotto-schemi in uno schema generale
6. **Analisi di qualità:**
   - Si verifica Correttezza, Completezza, Leggibilità e Minimalità dello schema concettuale

Cosa sono gli strumenti CASE? Descriverli.

Sono dei SW applicativi (Computer Aided Software Engineering) di ausilio alla ingegnerizzazione del software e forniscono un supporto a tutte le fasi principali dello sviluppo di una base di dati (Progettazione Concettuale, Logica e Fisica). Forniscono:

- Interfaccia grafica: modellazione dello schema in formato UML.
- Dizionario dei dati: memorizza informazioni dello schema.
- Strumenti integrati: eseguono automaticamente alcune fasi della progettazione.

ER-Win è un buon SW per fare queste cose. (Piattaforma Windows).
Taylor MDA e fabForce DBDesigner sono programmi equivalenti. (Piattaforma Linux).

Descrivi la progettazione logica.

La progettazione logica costituisce l'effettiva realizzazione della progettazione concettuale: durante questa fase si devono tener conto delle prestazioni dell'applicazione e quindi sono molto frequenti dei cambiamenti nella struttura e nello schema concettuale della applicazione:

- Ristrutturazione dello schema E-R: si basa sulla ottimizzazione e sulla semplificazione.
- Traduzione verso il modello logico: fa riferimento ad un modello logico (ad esempio il modello relazionale) e può incidere ancora sulla ottimizzazione in base la modello scelto. Si esegue anche la Normalizzazione.

Descrivere l'Analisi delle prestazioni su schemi E-R

Abbiamo alcuni parametri da controllare per verificare le prestazioni di uno schema:

- Costo di una operazione: numero medio delle occorrenze visitate per eseguire una operazione sul db. (Descritto nella Tavola delle Operazioni)
- Occupazione di memoria: numero di byte necessario per memorizzare i dati descritti dallo schema.
- Volume dei dati: numero di occorrenze di ogni entità ed associazione dello schema e dimensione di ciascun attributo. (Descritto nella Tavola dei Volumi)
- Caratteristiche delle operazioni: tipo di operazione, frequenza di richiesta, dati coinvolti. (Ognuna descritta opzionalmente negli Schemi di Operazione)

Cosa è la regola “ottantaventi” ?

È una regola convenzionale utilizzata per verificare l'efficenza di una base di dati, dove l'80% del carico è generato dal 20% delle richieste. Se la base di dati rispetta questa regola ha delle prestazioni adeguate.

Descrivere la ristrutturazione di uno schema E-R.

La fasi di ristrutturazione sono:

- Analisi delle ridondanze: si decide se mantenere o eliminare eventuali ridondanze nello schema.
- Eliminazione delle generalizzazioni: tutte le generalizzazioni vengono sostituite da altri costrutti.
- Partizionamento / accorpamento di entità ed associazioni: si accorpano o dividono alcuni concetti dello schema se necessario.
- Scelta degli identificatori primari: si seleziona un identificatore per quelle entità che ne hanno più di uno.

Descrivere la progettazione fisica della base di dati.

Utilizzando lo schema logico prodotto, le caratteristiche del sistema scelto e le previsioni sul carico applicativo viene prodotto lo schema fisico della base di dati (usando ad esempio le Create Table SQL su un server di Database).

Perché la Normalizzazione è necessaria su una base di dati ?

Perché decentralizza le informazioni aumentando anche inutilmente la ridondanza dei dati: ciò potrebbe causare delle incompatibilità sui dati, incoerenze, durante le fasi di aggiornamento dei dati o di cancellazione dei dati. Lo scopo della normalizzazione è quello di avere un dato in una sola locazione a cui eventuali altre entità sono collegate tramite delle relazioni.

Spiegare il concetto di dipendenza funzionale e fornire un esempio.

È un particolare vincolo di integrità per il modello relazionale che descrive i legami funzionali fra gli attributi di una relazione, come ad esempio lo Stipendio sull'entità Impiegato e si indica con: Impiegato -> Stipendio

Descrivi la forma normale di Boyce e Codd

La forma normale permette di eliminare i dati duplicati che sono contenuti in tabelle in cui non sono identificati con una chiave: solitamente l'utilizzo di chiavi aiuta nell'eliminare la ridondanza dei dati e a creare relazioni fra le varie tabelle. Queste relazioni sono in forma normale Boyce e Codd se per ogni dipendenza funzionale X -> A definita su di essa, X contiene e una chiave K di r, cioè X è superchiave per r. Anomalie e ridondanze vengono eliminate.

Su quale concetto si basa la decomposizione nella normalizzazione ?

Si basa sul concetto che se una relazione rappresenta più concetti indipendenti, la relazione deve essere decomposta in relazioni più piccole, una per ogni concetto (una per ogni dipendenza funzionale).

Descrivere le proprietà della decomposizione nella normalizzazione.

- Decomposizione senza perdita: quando dopo la decomposizione di una relazione r in 2, facendo il join fra le due si riottiene perfettamente la relazione r iniziale.
- Conservazione delle dipendenze: quando nella decomposizione ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti. Dire che una decomposizione che soddisfa questa proprietà vuol dire che conserva le dipendenze dello schema originario.
- Qualità delle decomposizioni: sono di qualità sufficiente quelle decomposizioni che soddisfano le precedenti 2 proprietà.

Parla della Terza Forma Normale.

Avendo la Forma Normale di Boyce e Codd la limitazione di non poter essere sempre utilizzata: si può allora utilizzare la Terza Forma Normale che può sempre essere applicata, permettendo sempre una Decomposizione senza Perdita e la Conservazione delle dipendenze ma non elimina completamente la ridondanza dei dati; la qualità della Terza Forma Normale è quindi inferiore a quella della Forma Normale di Boyce e Codd ma può essere sempre applicata.
Spesso in una decomposizione intesa ad ottenere una Terza Forma Normale si ottiene una Normalizzazione Boyce e Codd: ciò sta ad indicare come le due forme normali siamo simili.

Descrivere altre forme normali.

- **Prima Forma Normale**: stabilisce una condizione che sta alla base del modello relazionale stesso: gli attributi delle relazioni sono definiti su valori atomici e non su valori complessi quali insiemi o relazioni.
- **Seconda Forma Normale**: è una variazione debole della Terza Forma Normale in cui non sono definite dipendenze parziali cioè dipendenze fra un sottoinsieme proprio della chiave e altri attributi.
- Ne esistono anche altre ma quelle finora studiate danno una grande possibilità di giocare sulla Qualità e sulla Semplicità del sistema informativo.

Illustrare cosa vuol dire l' espressione: "Il valore Null è polimorfo", spiegando i significati che può assumere. Descrivere inoltre, il concetto di logica a tre valori, riportando le tabelle di verità dei connettivi logici AND, OR, NOT tenendo conto del valore nullo.

Il valore NULL può essere considerato polimorfo in quanto può assumere diversi significati:

- esiste ma non è noto
- non esiste
- non si sa

Introdotta in SQL-2 la logica a tre valori, stabilisce che, in caso di confronto con NULL, il valore restituito sarà *unknown* (e non *false* come in SQL-89). Le tabelle di verità dei connettivi logici AND, OR, NOT che considerano anche questa eventualità sono:

AND	V	U	F
V	V	U	F
U	U	U	F
F	F	F	F

OR	V	U	F
V	V	V	V
U	V	U	U
F	V	U	F

NOT	
V	F
U	U
F	V

Illustrare la proprietà di distribuzione della selezione e della proiezione rispetto agli operatori insiemistici indicando per quali operatori la proprietà vale fornendo, inoltre, un esempio di quando vale e di quando non vale.

La propreità di distribuzione o distributività, è quella proprietà che permette di stabilire un' equivalenza fra una selezione (o proiezione se si tratta dell' unione) applicata ad un' espressione e la stessa applicata ai singoli termini.
Tale proprietà può essere applicata all' unione U e alla differenza (da cui deriva la distributività rispetto alla selezione anche per ∩). Come già detto prima, la distributività rispetto alla proiezione vale solo per l' unione.
Un semplice esempio può essere:

$$\delta_{Cognome='Verdi'}(IMPIEGATI \cup DIRIGENTI) \equiv \delta_{Cognome='Verdi'} IMPIEGATI \cup \delta_{Cognome='Verdi'} DIRIGENTI$$

Un' esempio invece, che esprime quando non è possibile applicare tale proprietà è:

$$\prod_{Cognome}(IMPIEGATI \cap DIRIGENTI) \equiv \prod_{Cognome} IMPIEGATI \cap \prod_{Cognome} DIRIGENTI$$

Nell'ambito del modello relazionale, dire cosa si intende per vincolo di integrità referenziale, spiegare quali controllivengono eseguiti dal DBMS per verificarne la violazione e il soddisfacimento. Infine, elencare e descrivere le politiche di reazione che possono essere associate al vincolo di integrità referenziale in SQL.

Nell' ambito del modello relzionale il vincolo di integrità referenziale stabilisce che, se ho 2 relazioni la cui chiave primaria della prima compare nella seconda, ogni tupla della seconda dovrà avere ua corrispondenza nella prima (o al limite avere tale campo inizializzato a NULL).
Il DBMS andrà a controllare tali vincoli verificando che le condizioni espresse da asserzioni e costrutti check siano soddisfatte. Tale controllo avverrà ad ogni operazione o alla fine della transazione a seconda dell' impostazione (`immediate` o `deferred`) associata al vincolo.
Se tali controlli daranno esito positivo, cioè se i vincoli vengono violati, il DBMS attiverà delle opportune polithce di rezione precedentemente stabilite.
Queste politiche possono essere associate a una modifica o a una cancellazione e stabiliscono se:

- l' azione verrà propagata (`cascade`)
- il valore pendente verrà impostato a NULL (`set null`)
- il valore pendente verrà impostato al valore di default (`set default`)
- impedisce l' azione (`no action`)

Con riferimento a SQL, spiegare cosa si intende per variabile, indicando perchè servono e come vengono utilizzate. Fornire, inoltre, un esempio.

In SQL, le variabili vengono utilizzate per poter effettuare delle query altrimenti impossibili come quelle in cui si incorciano tuple di una stessa tabella.
Le variabili, consentono infatti di creare un alias per le tabelle da interrogare che si potrà utilizzare anche per svolgere delle query nidificate.
Un' esempio di utilizzo di variabili può essere il seguente:

```
select C1.Nome, C1.Cognome, C2.Nome, C2.Cognome
from CITTADINI as C1 inner join CITTADINI as C2
on C1.CFConiuge=C2.CF
where CFConiuge<>NULL and Sesso=”M”
```

Con riferimento a SQL, spiegare la differenza fra: inner join, left-outer join e right-outer join, spiegando perch´ ciascuno di essi serve ed illustrandone un esempio.

L' operazione di *inner join* si differenzia dalla *left outer join* e dalla *right outer join* in quanto non mantiene le tuple senza una corrispondenza.
Queste ultime 2 varianti della join infatti, incrociano le tuple dopodichè mantengono anche quelle senza corrispondenza a seconda dell' orientamento della join, ovvero se ad esempio eseguo una left outer join verranno mantenute le tuple senza corrispondenza della tabella a sinistra della join impostando a NULL i campi pendenti.

Data una interrogazione SQL che includa le clausole SELECT, FROM, WHERE, GROUP BY, HAVING e ORDER BY dire come viene eseguita (specificare come ed in che ordine vengono eseguite le clausole). Illustrare anche un esempio di query e il suo processo di esecuzione da parte del DBMS.

Le clausole presenti nell' interrogazione vengono eseguite nel seguente ordine: FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY. [da completare]