

## **MODELLO RELAZIONALE**

### **1) Cos'è l'integrità referenziale, quando può essere violata, e con quali regole si può gestire:**

Informazioni in relazioni diverse sono correlate attraverso valori comuni, può essere violata con tutte le operazioni di aggiornamento (inserimento, cancellazione, modifica).

Si può gestire con: -BLOCCARE L'ESECUZIONE CON MESSAGGIO D'ERRORE

- GESTIRE CON AZIONI COMPENSATIVE: es. nell'eliminazione :
  - rifiuto operazione (ins, canc, mod)
  - eliminazione in cascata (canc, mod)
  - introduzione valori nulli (canc, mod)

### **2) Cos'è un vincolo di integrità referenziale?**

E' un vincolo INTERRELAZIONALE. Un vincolo di integrità referenziale fra gli attributi X di una relazione R1 e un'altra relazione R2 impone ai valori su X in R1 di comparire come valori della chiave primaria di R2.

### **3) Cos'è una chiave?**

INTUITIVAMENTE: E' l'insieme minimo di attributi necessari ad individuare univocamente ogni n-upla della Relazione.

FORMALMENTE: E' la superchiave di dimensione minima, dove una superchiave è un insieme di k attributi tali che, se due n-uple (t1 e t2) sono distinte, esse avranno almeno un valore differente per uno di questi. NON ESISTE QUINDI dati  $t1 \neq t2 \rightarrow t1[k] = t2[k]$ .

### **4) Definizione chiave primaria?**

Chiave che NON AMMETTE VALORI NULLI.

### **5) Azioni che possono essere eseguite in risposta al fallimento di vincoli.**

Esistono 2 tipi di vincoli: INTRA RELAZIONALI (si viola se inserendo un valore nella tabella esterna, la sua chiave non è presente nella tabella), INTERRELAZIONALI (vincolo di integrità referenziale) T

o [1]BLOCCO OPERAZIONE RESTITUENDO ERRORE, O [2]INSERISCO NULL, O [3]ELIMINO IN CASCATA

### **6) Come risponde un DBMS al fallimento di vincoli di integrità referenziale?**

Può far fare una delle 3 cose sopra indicate. Cosa fare lo deve dire il progettista in fase di creazione della base di dati. (RESTRICT, CASCADE, SET NULL)

### **7) Cosa succede se violo altri vincoli (oltre al fatto che li rifiuta e richiede l'immissione)?**

Modifica: DBMS deve assicurarsi che il dominio sia corretto e che sia presente nella tupla DBMS spiega perchè non è stato accettato e può cercare di correggere (più che altro in cancellazione e modifica) la causa del rifiuto.

## **SQL**

### **8)Con quale costrutto si può memorizzare il risultato di una SELECT?**

Con INTO. SELECT->INTO->FROM...

### **9)Clausola WITH CHECK OPTION**

Tale clausola inserisce un controllo nelle viste aggiornabili: infatti se si cerca di aggiornare una vista che abbia tale clausola, l'operazione viene annullata qualora questa implicasse un inserimento/modifica dei dati per cui si abbia una violazione della condizione su cui tale vista è stata creata. Garantisce la consistenza della BASE DATI e con l'aggiunta di un CASCADED tali controlli possono essere estesi anche alle viste derivate. Con LOCAL no.

## **TRIGGER**

### **10)Cos'è un trigger?**

Un trigger è un costrutto che permette di stabilire azioni automatiche al susseguirsi di specifici eventi. Segue il modello ECA:

- Event: può essere un'operazione di Insert, Update, o Delete su un particolare attributo o su una relazione
- Condition: un'espressione booleana facoltativa, controllata dopo il verificarsi dell'evento e che se non soddisfatta non esegue l'action.
- Action: azione di risposta, di solito sequenza SQL o Transazione o Azione Esterna.

Non ammessi trigger ricorsivi.

### ***TRIGGER “EACH ROW” e di “STATEMENT”.***

I trigger possono essere definiti for “each row”, se così definiti hanno la peculiarità che la regola si ripete per ogni riga della tabella su cui è stato attivato (modifiche al dettaglio). Se non definito “for each row” allora è di default “STATEMENT”, la regola viene eseguita solo una volta per la tabella (modifiche grosse dimensioni). E' permesso l'uso di OLD e NEW(each row), altrimenti si usa trigger di istruzione(statement).

### ***TRIGGER BEFORE o AFTER.***

Varia il momento di esecuzione della regola rispetto all'evento, rispettivamente prima e dopo. Un metodo di scelta valido è valutare se la regola va a variare i dati oppure no.

### ***CONSTRAINT TRIGGER***

Si usa in presenza di schemi che sono in 3NF e non trasformabili in BCNF per decomporre lo schema in modo da raggiungere la BCNF perdendo una dipendenza funzionale. Questa deve essere gestita attraverso l'aggiunta di un vincolo supplementare alla base di dati.

## ER

### 11) Cardinalità dell'entità con chiave esterna:

Se ho un'entità con chiave identificata esternamente, essa deve avere Cardinalità minima e massima di partecipazione alla Relazione, pari ad 1, per garantire l'Univocità della chiave. (se chiave esterna  $\rightarrow (1,1)$ ).

### 12) Perché si dice che una dipendenza funzionale generalizza il vincolo di chiave?

Per la definizione di chiave, data una relazione  $R(X)$  con chiave  $K$  esiste una dipendenza funzionale tra  $K$  ed un qualunque insieme di attributi dello schema di  $R$

Infatti non è possibile trovare due tuple distinte che abbiano ugual valore in  $K$  ...

### 13) Cos'è una ridondanza in ER?

Una ridondanza è una ripetizione di informazioni nella base di dati: per esempio dati esplicitati, quando si potevano ricavare da altri o dati duplicati di per sé. Le ridondanze sono nel mirino della normalizzazione in quanto comportano sprechi di memoria e difficoltà di aggiornamento dei dati (ma non sempre sono dannose, a volte velocizzano gli accessi.. lo capisco dall'analisi del carico applicativo).

Negli schemi ER si possono riscontrare 4 tipologie diverse di ridondanze:

- Attributi di una stessa entità
- Attributi derivati da attributi di altre entità
- Attributi generati dal conteggio (più in generale operatori aggregati)
- Relazioni derivabili da un altro insieme di relazioni

### 14) Come si risolvono le ridondanze?

Tali ridondanze non vanno eliminate semplicemente, ma è opportuno prima di fare una stima dei costi delle operazioni che si hanno in presenza e in assenza di data ridondanza. Infatti alcune di esse anche se occupano la memoria procurano un grosso risparmio in termini computazionali. Per tale valutazione dei costi si deve usare la tabella dei volumi, delle operazioni e degli accessi.

### 13) Che cos'è la tavola degli accessi?

La tavola degli accessi è una tabella che viene definita per ogni operazione eseguibile nel database: in particolare è molto importante definirla per quel 20% delle operazioni che si suppone generare l'80% del carico computazionale. Nella tavola degli accessi vengono specificati :

- i concetti che l'operazione analizza nella sua esecuzione;
- il tipo di costrutto (entità, relazione);
- num accessi := il numero di volte che l'operazione deve accedere a quel costrutto;
- il tipo accesso:= se l'accesso è in lettura o scrittura.

Tale tabella ci consente di fare una stima sul costo computazionale complessivo di ogni operazione. La tavola degli accessi, insieme alla tavola dei volumi e quella delle operazioni , compongono il carico applicativo il quale viene stimato durante la fase di Raccolta e Analisi dei requisiti del ciclo di vita di un sistema informativo. Tramite questo stimiamo , durante la Progettazione logica , gli indici di prestazione di un DB (costo operativo e occupazione in memoria) per poi valutare se eliminare o no determinate RIDONDANZE.

#### **14)Tabella dei volumi?**

La tabella dei volumi è una tabella che fa parte della documentazione di una base dati e ci informa sul volume dei dati: tramite le seguenti informazioni: CONCETTO-TIPO-VOLUME. Con volume che tiene conto del numero di *occorrenze di ogni entità*, numero di *occorrenze di ogni associazione* e *dimensione di ciascun attributo* sia di entità che di associazione.

#### **15)Calcolare la tavola accessi dato il volume dei dati?**

Le caratteristiche delle operazioni vengono rappresentate nella tavola delle operazioni:

- riporta, per ogni operazione, la frequenza prevista ed il tipo di operazione (interattiva o batch)
- Si costruisce una tavola degli accessi basata su uno schema di navigazione

- Lo schema di navigazione è la parte dello schema E/R interessata dall'operazione, estesa con delle frecce che indicano in che modo l'operazione "naviga" i dati

- Per ragionare su quali entità/associazioni sono interessate dall'operazione, si considera il principio di "duplicazione minima" di informazioni, ovvero:

- entità identificate esternamente inglobano le relative associazioni (che quindi "non contano")
- le associazioni inglobano gli identificatori delle entità partecipanti
- entità figlie inglobano l'identificatore dell'entità genitore

Per ogni entità e associazione interessata dall'operazione, riporta il numero di istanze interessate, e il tipo di accesso (L: lettura; S: scrittura)

- Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni
- Ad es.: in media ogni impiegato partecipa a  $6000/2000 = 3$  progetti)

Riassumendo, l'analisi di prestazione

richiede:

- Una tavola dei volumi
- Una tavola delle operazioni
- Una tavola degli accessi per ogni operazione

#### **16)GENERALIZZAZIONI:**

Uno dei mezzi che ci permette di incrementare il grado di astrazione: si ottiene attraverso un entità padre e una o più entità figlie. Un'occorrenza di un'entità figlia è anche occorrenza di una padre e gli attributi del padre sono anche attributi dei figli. I figli estendono le proprietà padre con attributi propri.

Una Generalizzazione può essere:

- **Totale** ( ogni occorrenza del padre è almeno un'occorrenza di un figlio) o **Parziale**( un'occ. del padre può non corrispondere a nessun figlio)
- **Esclusiva** ( occ.padre è al massimo di un solo figlio) o **Sovrapposta** ( ci possono essere più entità figlie corrispondenti all'occ del padre contemporaneamente)

#### **17)Una generalizzazione sovrapposta può essere trasformata in una esclusiva? Come?**

QUELLE **SOVRAPPOSTE** si possono trasformare in **ESCLUSIVE**, creando una nuova Entità rappresentante l'intersezione tra l'entità precedenti.

### 18) Come si traducono e vantaggi e svantaggi delle varie soluzioni:

Vi sono 3 modi per tradurle quando passiamo al modello Relazionale :

1. **entità figlie raggruppate in padre:** potrebbe portare a tanti valori nulli sprecando memoria, ma nel caso in cui normalmente ci sono accessi a tutte le figlie in maniera pressochè uniforme è conveniente. ( viene aggiunto un attributo al padre per identificare il tipo di figlio e se un figlio aveva un relazione, questa diventa del padre, ma la cardinalità minima si abbassa a 0 in quanto ci potrebbero essere figli che non partecipano).
2. **entità padre divisa in quella di figlie:** è possibile farlo solo se la generalizzazione è TOTALE . E' utile usarlo quando so che ho molti accessi alle entità di una sola figlia, mi prende meno memoria in generale e non avrò valori nulli. Se partecipazione obbligatoria padre, ora è opzionale per le figlie.
3. traduzione di una generalizzazione in un' **associazione** , ovvero per ogni figlio viene aggiunta una relazione che identifica esternamente i figli col padre.. Conviene farlo quando la generalizzazione **NON** è totale.

### 19) Cos'è il vincolo di Unique ?

Serve a sottolineare che un particolare attributo può apparire nella relazione solo una volta (è unico). Quando andiamo a tradurre una relazione 1 a 1 da ER a Relazionale e decidiamo che un "entità ingloba l'altra nella relazione" , la chiave dell'entità inglobata è **unique** in quanto la cardinalità di partecipazione massima è 1. Si indica sottolineando in modo seghettato l'attributo .

## NORMALIZZAZIONI

### 20) Cos'è la normalizzazione

E' il processo attraverso il quale uno schema logico può essere trasformato per essere conforme ai requisiti delle FORME NORMALI che stabiliscono il grado di immunità rispetto a criticità.

### 21) Come si esegue la normalizzazione

Algoritmo (per portare in 3FN):

- 1- Si identifica una copertura minimale G per l'insieme F.
- 2- Partizionamento: si dividono le df in G gruppi con lo stesso determinante (attributi a sx della df)
- 3- Sintesi schemi: per ogni gruppo si costruisce uno schema usando tutti e solo gli attributi che compaiono in almeno una df del gruppo.
- 4- Decomposizione: se non esiste uno schema la cui chiave determina tutti gli attributi di R(X) si sintetizza una chiave K ( $K \rightarrow X$ ) ed un nuovo schema costituito dagli attributi di K.

### 22) Definizione Attributo Primo:

Un attributo è detto PRIMO se compare in qualche chiave della relazione.

### 23) Definizione di Dipendenza Completa:

Data una dipendenza funzionale  $A \rightarrow B$  su due insiemi di attributi A e B, si dice che B è **COMPLETAMENTE DIPENDENTE** da A se ogni possibile sottoinsieme di attributi di A non determina B.

### 24) Definizione 1FN:

Una tabella si dice in **prima forma normale** (1NF) se e solo se ciascun attributo è definito su un dominio con valori atomici, e non c'è più di una colonna riferita ad uno stesso attributo (non ci sono attributi aggregati o attributi multivalore).

## 25) Definizione 2FN:

Una Relazione è detta in 2NF se :

- è in 1NF
- Ogni attributo **non primo**, **dipende completamente** dalla Chiave.

## 26)3NF e BCNF.

Processo di standardizzazione che ultima la progettazione logica del db. Essa mi permette di evitare **le anomalie di cui più comunemente un db soffre : inserimento, aggiornamento e cancellazione**. Generalmente vengono eliminate le ridondanze, gli attributi estranei alle df e la relazione iniziale viene decomposta in piu tabelle. L'obiettivo minimo, e sempre raggiungibile, quando si normalizza un db è portarlo almeno in 3NF , così definita:

*“Una relazione  $R(X)$  si dice in **3nf** se è in **1NF** e per ogni dipendenza non banale  $X \rightarrow A$  ,si ha che o  $X$  è **superchiave** per  $R$  o  $A$  è **attributo primo** .”*

Sebbene riusciamo ad eliminare quasi tutte le anomalie , può non essere ancora immune all' **Anomalia di Cancellazione** . Per questo se possibile si prova ad alzare il grado di normaliz. portando  $R$  in BCNF , così che il db sia immune anche a quest'ultima. la Boyce Code normal Form è pertanto così definita:

*“Una relazione  $R(X)$  si dice in **BCnf** se è in **1NF** e per ogni dipendenza non banale  $X \rightarrow A$  ,si ha che o  $X$  è **superchiave** per  $R$  .”*

NB. Sebbene una  $R(x)$  in BCNF è anche in 3NF, non è sempre vero il contrario.

## 27)Dimostrare che se una tabella è in 3NF è anche in 2NF (Updated)

### **Dimostrazione per assurdo:**

Supponiamo di avere una tabella che è in 3NF, ma non è in 2NF. Sia  $A \rightarrow X$  una dipendenza funzionale che fa sì che la tabella non sia in 2NF:

quindi che abbia **A sottoinsieme della chiave** e dove **X non sia un attributo primo** .

Si verifica immediatamente l'assurdo in quanto se  $A$  è una parte della chiave (quindi non superchiave! ) e  $X$  è un attributo non primo la tabella non è nemmeno in 3NF ! **ASSURDO**

## 28)Come si normalizza una tabella ?

(si mette in 1NF, si prende la copertura minima e si decompone sulla base di quella fino a raggiungere la 3NF, o superiore)

## 29)Quando 2 insiemi di df sono equivalenti?

Quando gli attributi implicati logicamente dall'uno possono essere implicati anche dall'altro e viceversa, i.e. quando la chiusura è la medesima.

## 30)Cos'è la chiusura di un insieme di df?

Dato  $\langle R(X), F \rangle$ , l'insieme  $F^+$  di tutte le df implicate logicamente da quelle in  $F$  è detto CHIUSURA di  $F$

## 31)Come si determina l' implicazione logica di df?

L'implicazione logica si determina applicando gli **Assiomi di Armstrong** alle df:

- **Transitività** : se  $A \rightarrow B$  e  $B \rightarrow C$  allora  $A \rightarrow C$
- **Riflessività** : se  $B \subset A$  e  $B \rightarrow C$  allora  $A \rightarrow C$
- : se  $A \rightarrow B$  allora  $AC \rightarrow BC$

### 32)Cos'è la copertura minima ?

E' l'insieme minimale di df che identificano la medesima Chiusura.

Si calcola individuando:

- df ridondanti
- attributi estranei alle df

### 33)Cos'è una decomposizione senza perdite ? Quale è la condizione per evitare perdite nelle decomp.?

“Sia data una relazione  $\langle R(x), F \rangle$ , si definisce la sua **decomposizione**  $R1(A)$  e  $R2(B)$ , dove  $A$  e  $B$  sono sottoinsiemi di  $X$ , tali che  $A \cup B = X$ .”

Una decomposizione si dice **senza perdita** se è possibile con un ( lossless) join tra  $R(A)$  e  $R(B)$  ottenere  $R(X)$ .

Ne consegue che deve esser un insieme  $C = (A \text{ intersecato } B)$  che è la **superchiave** di almeno una delle due relazioni, i.e.  $C \rightarrow A$  o  $C \rightarrow B$ . (condiz.sufficiente, - non necessaria)

### 34)Condizione per cui una decomposizione conserva le dipfunz

Brutalmente si dice che una decomposizione conserva le dipendenze funzionali quando non vanno perse, ovvero, data una df, tutti gli attributi che ne partecipano sono nella medesima relaz. in seguito alla decomposizione.

Matematicamente si traduce in :

$FX \rightarrow FA \cup FB$

## TRANSAZIONI

### 35)Cos'è una transazione?

Unità logica di elaborazione su database che include una o più operazioni di accesso.

Nell'esecuzione della transazione il DBMS è responsabile del fatto che:

- tutte le operazioni della transazione siano completate con successo
- nessuna operazione della transazione abbia effetto sul database

### 36)Cosa vuol dire che una transazione è ben formata?

Una transazione è ben formata se:

- Inizia con un begin transaction.
- Termina con un end transaction.
- La sua esecuzione comporta il raggiungimento di un commit o di un rollback work e dopo il commit/rollback non si eseguono altri accessi alla base di dati.

Esempio di transazione ben formata

*begin transaction;*

*update CONTO set saldo = saldo - 1200*

*where filiale = '005' and numero = 15;*

*update CONTO set saldo = saldo + 1200*

*where filiale = '005' and numero = 105;*

*commit work;*

*end transaction;*

### 37)Casi in cui una Transazione termina :

rottura di vincoli immediati (vengono verificati nello stato partially committed e sono gestiti dal DBMS effettuando il rollback)

rottura di vincoli differiti ( vengono verificati nello stato COMMITTED, credo si gestisca come i vincoli immediati)

- 1)Guasto sistema(memoria),
- 2)Errore sistema(valori errati),

- 3) Errori Locali o eccezioni
- 4) Attuazione controllo concorrenza
- FIN QUI SI PUO' RIPRISTINARE
- 5) Errore sul disco //raro
- 6) problemi catastrofici //raro

### **38) Motivi per cui può terminare una transazione in Oracle:**

*Penso che si riferisse al fatto che si può chiudere per commit/rollback, per chiusura della sessione da parte dell'utente (e allora è commit), da parte del super-user (allora rollback).*

IN CASO DI ERRORE in una qualunque istruzione SQL, l'intera transazione usa ROLLBACK.

### **39) Dopo che ho dato il commit per chiudere una transazione sono sicuro che tutto è andato a buon fine oppure no?**

La risposta giusta era NO perchè ci sono alcuni vincoli (detti vincoli differiti) che possono essere definiti in modo che la loro verifica avvenga solo dopo il commit (e quindi la verifica potrebbe causare il fallimento della transazione)

### **40) Tipi di vincoli**

- **Vincolo differito (deferred):** Si differisce dall'altro perchè può essere temporaneamente ignorato all'interno di una transazione, ciò consente ad esempio l'inserimento di occorrenze con attributi che violano un vincolo di integrità referenziale seguito dalla modifica per risolvere lo stato inconsistente dei dati. Se non specificato un vincolo è di default IMMEDIATO, sennò alla creazione → deferred. Sono verificati solo al termine della transazione. Una loro violazione viene gestita dal DBMS che impone un ROLLBACK della transazione.

VANTAGGI: azioni più sofisticate e aggiornamenti più dinamici alla base di dati.

SVANTAGGIO: queste azioni se violano i vincoli occorre fare una ROLLBACK su tutto.

- **Vincolo immediato :** generano un errore appena viene eseguita un'istruzione che causa la loro violazione. Può essere gestito e intercettato dalla transazione (CATCH).

*(consistenza nelle transazioni)*

### **41) Le proprietà "acide" delle transazioni ed in particolare quella atomica :**

**Atomicità, Consistenza, Isolamento, Persistenza** (ACID -- Atomicity, Consistency, Isolation, Durability)

la proprietà **atomica** richiede che ogni insieme di istruzioni venga trattato come un insieme indivisibile. Nel caso di operazioni avvenute su db il DBMS può confermare la conclusione dell'operazione attraverso "commit". Oppure potrebbe capitare che avvenga un errore oppure che venga richiesta l'annullamento di tutte le operazioni avvenute fino a quel momento. questi due ultimi scenari riguardano il rollback.

### **42)livelli di isolamento fra le transazioni**

L'Isolamento è imposto dal DBMS. Se ciascuna transazione non rende visibili i propri aggiornamenti ad altre transazioni finchè non viene confermata, si impone una forma di isolamento risolutiva del problema delle letture sporche e degli annullamenti (ROLLBACK) a cascata.

**LIVELLO ISOLAMENTO 0 (READ UNCOMMITTED):** non sovrascrive le letture sporche di transazioni di livello più alto

**LIVELLO ISOLAMENTO 1 (READ COMMITTED):** non ha perdita di aggiornamento elimina la dirty read (prima anomalia di isolamento).

**LIVELLO ISOLAMENTO 2 (READ REPEATABLE COMMITTED):** non ha perdita di aggiornamento e nessuna lettura sporca

**LIVELLO ISOLAMENTO 3 (SERIALIZABLE):** liv.2 + letture ripetibili. (isolamento vero). Se anche qui ho anomalie si usa un ROLLBACK come DEADLOCK per tornare allo stato in cui non ho anomalie.



## **ANOMALIE DI ISOLAMENTO**

### **43)Cos'è il dirty-read**

Considerando 2 transazioni T1 e T2 che operano simultaneamente dove la prima effettua una lettura del dato A mentre la seconda un aggiornamento sul dato A. Si ha una DIRTY READ nel caso in cui il risultato di T1 sia influenzato dall'azione di T2 anche se T2 non ha effettuato la COMMIT.

### **44)Cos è il phantom read**

Siamo sempre nella casistica di due transazioni concorrenti dove T1 effettua un conteggio può restituire 2 risultati diversi a causa dell'aggiunta di occorrenze ad opera di T2.

### **45)Cos è il non repeateble read**

Stessa casistica della DIRTY READ solo che qui la seconda transazione ha già effettuato la commit quindi non è più possibile un ROLLBACK per l'annullamento degli effetti, si ricorda che essendo 2 transazioni simultanee non devono influenzarsi a vicenda.

### **46)Differenza tra nonrepeatable read e dirty read:**

*(nel dirty read si legge il dato prima del commit,nella prima è stato dato invece)*

Entrambi sono la conseguenza di mancanza di isolamento tra i processi. In dirty T1 legge un dato modificato da T2 prima che T2 l'abbia confermato invece in non reap. legge un dato modificato su cui T2 ha già dato commit

### **47)cursore**

```
DECLARE nome_cursore CURSOR FOR query;
```

un cursore dei database è una struttura che permette di scorrere i record restituiti da una query. Essi possono essere di sola lettura o, se l'implementazione lo consente, possono essere usati per modificare o cancellare le righe

## **MODELLO RELAZIONALE**

- 1) Cos'è l'integrità referenziale, quando può essere violata, e con quali regole si può gestire:
- 2) Cos'è un vincolo di integrità referenziale?
- 3) Cos'è una chiave
- 4) Definizione chiave primaria?
- 5) Azioni che possono essere eseguite in risposta al fallimento di vincoli.
- 6) Come risponde un DBMS al fallimento di vincoli di integrità referenziale?
- 7) Cosa succede se violo altri vincoli (oltre al fatto che li rifiuta e richiede l'immissione)?
- 8) Con quale costrutto si può memorizzare il risultato di una SELECT?
- 9) Clausola WITH CHECK OPTION

## **TRIGGER**

- 10) Cos'è un trigger?
- TRIGGER "EACH ROW" e di "STATEMENT".  
TRIGGER BEFORE o AFTER.  
CONSTRAINT TRIGGERER
- 11) Cardinalità dell'entità con chiave esterna:
  - 12) Perché si dice che una dipendenza funzionale generalizza il vincolo di chiave?
  - 13) Cos'è una ridondanza in ER?
  - 14) Come si risolvono le ridondanze?
  - 15) Che cos'è la tavola degli accessi?
  - 16) Tabella dei volumi?
  - 17) Calcolare la tavola accessi dato il volume dei dati?

## **18) GENERALIZZAZIONI:**

- 19) Una generalizzazione sovrapposta può essere trasformata in una esclusiva? Come?
- 20) Come si traducono i vantaggi e svantaggi delle varie soluzioni:
- 21) Cos'è il vincolo di Unique?

## **NORMALIZZAZIONI**

- 22) Cos'è la normalizzazione
- 23) Come si esegue la normalizzazione
- 24) Definizione Attributo Primo:
- 25) Definizione di Dipendenza Completa:
- 26) Definizione 1FN:
- 27) Definizione 2FN:
- 28) 3NF e BCNF.
- 29) Dimostrare che se una tabella è in 3NF è anche in 2NF (Updated)
- 30) Come si normalizza una tabella?
- 31) Quando 2 insiemi di df sono equivalenti?
- 32) Cos'è la chiusura di un insieme di df?
- 33) Come si determina l'implicazione logica di df?
- 34) Cos'è la copertura minima?
- 35) Cos'è una decomposizione senza perdite? Quale è la condizione per evitare perdite nelle decomp.?
- 36) Condizione per cui una decomposizione conserva le dipfunz

## **TRANSAZIONI**

- 37) Cos'è una transazione?
  - 38) Cosa vuol dire che una transazione è ben formata?
  - 39) Casi in cui una Transazione termina:
  - 40) Motivi per cui può terminare una transazione in Oracle:
  - 41) Dopo che ho dato il commit per chiudere una transazione sono sicuro che tutto è andato a buon fine oppure no?
  - 42) Tipi di vincoli
  - 43) Le proprietà "acide" delle transazioni ed in particolare quella atomica;
  - 44) Livelli di isolamento fra le transazioni
- ANOMALIE DI ISOLAMENTO
- 45) Cos'è il dirty-read
  - 46) Cos'è il phantom read
  - 47) Cos'è il non repeatable read
  - 48) Differenza tra nonrepeatable read e dirty read:

- 49) cursore