1.Dare la definizione formale di chiave / chiave primaria:/ chiave primaria modello relazionale:

Una chiave di una relazione è un insieme di attributi che distingue fra loro le tuple della

Relazione. Def. formale: Sia S(A1, ...,An) uno schema di relazione. Un insieme X ⊆ US è chiave di S se verifica entrambe le seguenti proprietà: 1. Qualsiasi sia lo stato di S, non esistono due tuple distinte di S che abbiano lo stesso valore per tutti gli attributi in X 2. Nessun sottoinsieme proprio di X verifica la proprietà (1). Un insieme di attributi che verifica la proprietà (1) ma non la proprietà (2), è detto super-chiave.

3.Dare la definizione formale di chiave esterna nel modello relazionale:

Siano: S ed S′ due relazioni. Y ⊆ US’ una chiave per S′. X ⊆ US un insieme di attributi di S tale che Y e X contengano lo stesso numero di attributi e di dominio compatibile X è una chiave esterna di S su S′ se, qualsiasi siano gli stati di S ed S′, per ogni tupla t di S esiste una tupla t′ di S′ tale che t[X] = t′[Y ]. S viene detta relazione referente. S′ viene detta relazione riferita

dare la definizione formale di relazione nel modello relazionale e spiegarecome il modello relazionale organizza i dati:

una relazione su D1, D2, …, Dk è un sottoinsieme finito del prodotto cartesiano D1 x D2 x … x Dk. Il modello relazionale organizza i dati in relazioni, cioè tabelle, con righe chiamate tuple e colonne contenenti gli attributi di ciascuna tupla. Una relazione,

sottoinsieme del prodotto cartesiano di k domini, ha grado k. Ogni tupla di una relazione di grado k ha k componenti, una per ogni dominio su cui è definita la relazione cui la tupla appartiene. La cardinalità di una relazione è il numero di tuple appartenenti alla relazione

4.Spiegare cosa sono DDL, DML e SDL ed illustrare i principali comandi da essi forniti:

): Il DDL (Data Definition Language) permette di specificare e modificare lo schema della

base di dati, e i vincoli di integrità, riguarda il livello logico e quello esterno. Il DML (Data Manupulation Language) permette di effettuare query o interrogazioni riguardanti i dati della base di dati, riguarda il livello logico e le viste. LʼSDL (Storage Definition Language) definisce lo schema fisico del database e riguarda solo il livello fisico. I comandi di ddl sono create, alter, drop. Quelli di dml sono insert, update,delete,select. Quelli di sdl sono sono quelli inerente al engine memory

5.Spiegare la differenza tra schema e istanza di una base di dati e quali sono i principali comandi SQL per la loro gestione:

Entrambi sono definiti nel DBMS indipendentemente dal modello dei dati.

• Schema (logico) della base di dati: descrizione dei dati, specificata tramite il modello dei dati, cambia raramente (Es. l’insieme di tabelle, colonne, tipi di dati, vincoli, indici e relazioni tra tabelle) . I suoi comandi sono quelli del DDL.

•Istanza della base di dati: L’istanza di una base di dati rappresenta lo stato attuale dei dati presenti nel database, ovvero i record memorizzati nelle tabelle in un dato momento. È l’insieme dei dati che cambia nel tempo a seguito di operazioni di inserimento, modifica o cancellazione. I suoi comandi sono quelli del DML.

12.Illustrare le principali tipologie di soluzioni per sviluppare una applicazione che si interfaccia a una base di dati relazionale e vantaggi e svantaggi/ forza o debolezza

siccome l'utilizzo esclusivo di SQL presenta dei limiti occorre integrarlo con un linguaggio di programmazione, esistono 2 principali integrazioni : integrare SQL con un linguaggio esistente, questo approccio è detto client- side utilizzo di estensioni procedurali di SQL,approccio server-side. Per il primo caso il codice si esegue all'esterno del DBMS, il linguaggio con cui si integra SQL viene chiamato linguaggio ospite. Per l'integrazione tra SQL e linguaggio ospite si utilizzano delle API che fanno da interfaccia tra le due parti. Un altro modo è che i comandi SQL sono utilizzati direttamente insieme al linguaggio ospite, un pre-compilatore tradurrà il mix dei 2 linguaggi in linguaggio ospite, sostituendo i comandi SQL con delle chiamate al DBMS. il secondo modo si chiama embedded SQL. La seconda integrazione prevede l'aggiunta di costrutti dei linguaggi di programmazione procedurali a SQL. Ogni DBMS ha una sua estensione di costrutti procedurali. Per concludere con il primo approccio, trovandoci all'esterno del DBMS avremo più indipendenza da quest'ultimo, lo svantaggio di ciò è la comunicazione tra l'applicazione e il DBMS richiede tempo e costo, approccio ideale se si hanno richieste semplici per il DBMS in modo che l'interazione costi poco. Il secondo approccio presenta dipendenza diretta dal DBMS utilizzatoo poichè i costrutti sono forniti dal DBMS, maggiore efficienza in quanto le istruzione SQL le eseguiamo direttamente dove ci troviamo (DBMS) ma minore espressività, come già anticipato SQL e le estensioni non sono complete per tutte le necessità.Spesso si usa una combinazione tra l'approccio client-side e quello server-side in modo da sfruttare i vantaggi.

9. Spiegare le attività che si svolgono nella fase di progettazione logica di una base di dati, evidenziando anche l’input ed output di questa fase:

- RACCOLTA ED ANALISI DEI REQUISITI, dove vengono definite le caratteristiche della base di dati, condotta in modo informale quindi con interviste e sondaggi. L’output è un documento in linguaggio naturale di specifica dei requisiti, che possono essere diversi (informative, sui vincoli di integrità ed autorizzazione, sulla popolosità della base di dati e sulle operazioni);

- PROGETTAZIONE CONCETTUALE, dove viene creato uno schema concettuale, quindi una

descrizione formale e ad alto livello del contenuto della base di dati. Esso è indipendente

dall’implementazione della base di dati ed è definito tramite un opportuno modello

concettuale. In questa fase inoltre vengono fatte delle verifiche di qualità che possono portare ad un ridisegno dello schema generato. Inoltre, vi è una documentazione a corredo dello schema con i vincoli di integrità non rappresentabili nello schema, scelte progettuali e altre informazioni;

- PROGETTAZIONE LOGICA, ovvero la traduzione dello schema concettuale. L’output è lo

schema logico nel DDL del DBMS target e vengono considerati l’integrità, la sicurezza e

l’efficienza;

- NORMALIZZAZIONE, dove si verifica la qualità dello schema logico prodotto e viene effettuata tramite degli strumenti formali. Prende questo nome nel caso di basi di dati relazionali e tale attività può portare ad una ristrutturazione dello schema logico prodotto;

- PROGETTAZIONE FISICA, dove vengono effettuate alcune scelte sulla memorizzazione fisica dei dati e l’output è lo schema fisico che descrive le strutture di memorizzazione ed accesso dei dati.

10.Spiegare cosa sono le proprietà ACIDE e, per ognuna di esse, illustrare le garanzie che offre ai dati.

• **Consistency** (consistenza);   Una transazione deve agire sulla base di dati in modo corretto. garantito dal sottosistema di controllo della concorrenza,

•**Isolation** (isolamento);

L’esito di una transazione non deve essere influenzato dall’esecuzione contemporanea di altre transazioni.  assicurata dal sottosistema di controllo della concorrenza

• **Atomicity** (atomicità);

Le operazioni di una transazione devono essere trattate come una singola unità.  assicurata dal sottosistema di ripristino (recovery) del DBMS.

• **Durability** (persistenza, o durabilità).

I risultati di una transazione che ha effettuato il commit devono essere resi permanenti nella base di dati. assicurata dal sottosistema di ripristino

Spiegare cosa è JDBC e principali comandi offerti:

JDBC (Java Database Connectivity) è un'API (Application Programming Interface) che consente alle applicazioni Java di interagire con i database. Essenzialmente, fornisce un insieme di classi e metodi che consentono agli sviluppatori di scrivere codice Java per connettersi a un database, eseguire query SQL, aggiornare i dati e gestire le transazioni.

Comandi:

• generico (per SQL statico e dinamico);

Per l’esecuzione di un comando generico, si usa un oggetto di classe Statement, che viene creato dal metodo createStatement della classe Connection

• preparato (solo per SQL statico);

Un comando SQL “preparato” viene compilato una sola volta e poi può essere eseguito più volte.

• callable statement (invocazione di procedure server-side).

17. Spiegare cos’è un DBMS ed i principali servizi da lui offerti e cosa lo rende migliore rispetto al file system

un DBMS (Data Base Management System) è un sistema software che fornisce gli strumenti per la gestione delle informazioni. Le sue caratteristiche sono : (1) gestione di dati in memoria secondaria, in quanto il data base può avere dimensioni maggiori rispetto alla capacità della memoria centrale; (2) Condivisione dei dati tra applicazioni e utenti diversi, con riduzione della ridondanza dei dati, delle inconsistenze tra dati e con un meccanismo di controllo dell'accesso concorrente; (3) Persistenza dei dati, il tempo di vita del dato non è limitato a quello dell'esecuzione dei programmi che li utilizzano; (4) Affidabilità dei dati grazie a delle funzionalità di backup e ripristino; (5) Sicurezza dei dati, dovuta a dei meccanismi di autorizzazione, si stabiliscono le regole a livello centralizzato; (6) Efficacienza, le varie operazioni sfruttano un insieme di risorse accettabile per gli utenti; (7) Efficacia, capacità di rendere produttive le attività degli utenti. Il DBMS è meglio perché: riesce a gestire grandi quantità di dati (evitando ridondanze e inconsistenze) e si interpone tra le applicazioni e il file system, fornendo un accesso centralizzato ai dati, astraendo i dettagli relativi alla gestione dei file.

13.Spiegare l’architettura a 3 livelli di un BDMS:

(1) livello logico: a questo livello è definito lo schema logico. L'astrazione fornita da questo livello facilità l'uso della base di dati da parte degli utenti finali; (2) Livello fisico: è il livello più basso, in cui viene definito lo schema fisico della base di dati, che specifica come i dati nello schema logico vengono effettivamente memorizzati tramite strutture di memorizzazione. A questo livello si ha la possibilità di ottimizzare lo storage e la ricerca dei dati (ma quest'operazione viene solitamente effettuata dall'amministrazione della base di dati, quindi l'utente finale non ha bisogno di interagire con il livello fisico); Questi due livelli sono sempre presenti. Invece, in alcune basi di dati, solitamente quelle con schemi medio-grandi, si ha un'ulteriore astrazione sopra al al livello logico: il livello esterno o livello delle viste. Esso permette di definire delle viste, ciascuna delle quali consente di "vedere" solo una porzione dell'intero schema logico. Ciò è utile, ad esempio, per controllare quali dati sono accessibili da varie applicazione, creando per ciascuna di esse una vista limitata ai dati che deve avere a disposizione.

14. Spiegare il concetto di indipendenza logica e fisica di un DBMS:

Indipendenza fisica Si riferisce alla separazione tra il modo in cui i dati sono memorizzati fisicamente e la logica delle query o delle applicazioni che accedono a tali dati. Consente di modificare la struttura fisica del database senza influenzare la logica delle applicazioni che lo utilizzano

Indipendenza logicaSi riferisce alla separazione tra la vista dei dati e il modo in cui vengono memorizzati fisicamente. In altre parole, gli utenti possono accedere e manipolare i dati senza dover preoccuparsi della struttura di archiviazione sottostante o di come vengono effettivamente memorizzati nel database.

15.Spiegare cos’è una transazione ed illustrare le proprietà che il DBMS assicura alla sua esecuzione e i moduli proposti per la verifica di tali proprietà

): la transazione è una sequenza di operazioni sul DB cui si vogliono associare particolari caratteristiche di correttezza, robustezza, isolamento anche in presenta di accessi concorrenti (es: bonifico bancario, acquisto biglietto, prenotazione aerea, etc..). Sistema transazionale: mette a disposizione un meccanismo per la definizione e l'esecuzione di transazione. Una transazione è un'unità logica di lavoro, non ulteriormente scomponibile, e può concludersi con un successo o un insuccesso: in caso di successo, il risultato delle operazioni eseguite deve essere memorizzato in modo permanente nella base di dati in caso di insuccesso, la base di dati deve tornare allo stato precedente l'inizio della transazione. Il DBMS assicura le seguenti proprietà all'insieme di operazioni che costituiscono una transazione: Atomicity - atomicità, Consistency - consistenza, Isolation *( “scrivi anche una micro definizione di cosa sono”)*

6.Spiegare come il DBMS gestisce i valori nulli e cosa sono:

SQL usa una logica a tre valori per valutare il valore di verità di un predicato SQL: TRUE (T) FALSE (F)UNKNOWN (?) UNKNOWN indica che il valore di verità di un predicato valutato su una data tupla non è determinabile. Un predicato semplice valutato su un attributo a valore nullo da come risultato della valutazione "?" Il valore di verità di un predicato complesso viene calcolato in base a queste tabelle di verità. Nelle espressioni (ad es. aritmetiche) se un argomento è NULL allora il valore dell'intera espressione è NULL NB: se e1 ed e2 sono NULL, e1=e2 non è vero, è UNKNOWN Una tupla per il cui valore di verità è ? non viene restituita dall'interrogazione Viceversa, in un vincolo d'initegrità se la valutazione della condizione di controllo restituisce ? il vincolo non è violato.

7.Dare la definizione/ cosa sono l’identificatore nel modello ER illustrandone le varie tipologie:

ʼidentificatore è un insieme di attributi e/o entità che identificano univocamente le istanze dellʼentità. Un identificatore è minimale qualsiasi sottoinsieme proprio non è un identificatore. Durante la progettazione concettuale per ogni entità è bene identificare tutti gli identificatori minimali. Gli identificatori hanno senso solo per le entità e non per le associazioni: nellʼinsieme di istanze di unʼassociazione si hanno tutte le tuple distinte: non cʼè bisogno di identificatori. A volte non è possibile identificare unʼistanza di entità solo sulla base dei suoi attributi, cioè due istanze diverse possono coincidere su tutti gli attributi. Si utilizza allora il fatto che tale istanza partecipi ad una particolare istanza di associazione con una data istanza di unʼaltra entità. Lʼentità identificata in questo modo viene detta entità debole Tipologie di identificatori: Interni: uno o più attributi dellʼentità Esterni: una o più entità collegate da una associazione allʼentità a cui si riferiscono (identificazione esterna da tale entità attraverso tale associazione) Misti: combinazioni delle due tipologie precedenti Semplici: un solo elemento Composti: più di un elemento Le entità deboli hanno sempre cardinalità (1, 1) rispetto allʼassociazione attraverso cui avviene lʼidentificazione.

11.Spiegare cos’è l’integrità referenziale e il suo comando per la gestione in SQL

Il vincolo di integrità imposto dalla definizione di chiave esterna (cioè che i valori assunti da una chiave esterna nella relazione referente devono esistere come valori della chiave primaria nella relazione riferita) è chiamato vincolo di integrità referenziale. Qualsiasi siano gli stati di S ed S′, per ogni tupla t di S esiste una tupla t′ di S′ tale che t[X] = t′[Y ]. In pratica, verificando che tale vincolo sia soddisfatto, il DBMS garantisce che le associazioni rappresentate dalle chiavi esterne siano coerenti. L'integrità referenziale può essere violata da: inserimenti e modifiche (del valore della chiave esterne) nella relazione referenziale oppure cancellazioni e modifiche (del valore della chiave primaria) nella relazione riferita. La clausola opzionale ON DELETE permette di specificare le azioni da eseguire quando vengono cancellate delle tuple nella tabella riferita. Analogamente, la clausola opzionale. ON UPDATE permette di specificare come gestire le modifiche dei valori della chiave primaria nella tabella riferita. Per entrambe le clausole, le opzioni disponibili sono: NO ACTION : la cancellazione / modifica della tupla nella tabella viene eseguita solo se non ci sono riferimenti a essa nella tabella referente, altrimenti il DBMS rifiuta l'operazione e segnala un errore. CASCADE : la cancellazione / modifica della tupla nella tabella riferita implica anche la cancellazione / modifica di tutte le tuple nella tabella referente che fanno riferimento a essa.SET NULL : la tupla nella tabella riferita viene cancellata / modificata e i valori della chiave esterna che facevano riferimento a essa vengono posti a NULL (se ammesso). SET DEFAULT : la tupla nella tabella riferita viene cancellata / modificata, e i valori della chiave esterna che facevano riferimento a essa vengono posti al valore di default (se specificato nella creazione della tabella). L'opzione di default è NO ACTION . Osservazione: queste opzioni sono specificate nella creazione della tabella referente, anche se riguardano operazioni effettuate sulla tabella riferita. In quest modo, se ci sono più tabelle referenti per una stessa tabella riferita, ciascuna di esse può specificare comportamenti diversi, in base alle esigenze del dominio applicativo.

spiegare le varie possibilità per tradurre una relazione/associazione uno a uno nel modello relazionale:

sono possibili più modalità di traduzione: dipende dal valore della cardinalità minima:

Se solo una delle due entità ha partecipazione obbligatoria si aggiungono alla sua relazione lʼidentificatore dellʼaltra entità e gli attributi propri dellʼassociazione. Se entrambe le entità hanno partecipazione obbligatoria, lʼassociazione può essere tradotta mediante lʼaggiunta di attributi a una qualsiasi delle due entità. Infine se entrambe le entità hanno partecipazione opzionale, lʼaggiunta di attributi che devono per forza essere opzionali, introduce inevitabilmente dei valori nulli. (come sopra ma con aggiunta di circoletto). Le associazioni unarie si traducono in modo analogo, considerando i ruoli

illustrare come si traduce una associazione uno a molti nel modello ER

allʼinterno di uno schema relazionale illustrando tutti i possibili casi : Si può tradurre in due modi: tramite lʼaggiunta di attributi → aggiungere alla relazione dal lato “unoˮ lʼidentificatore dellʼaltra entità associata e eventuali attributi dellʼassociazione

tramite una nuova relazione che contiene: identificatori delle due entità partecipanti, che fanno da chiavi esterne e attributi propri dellʼassociazione.

Illustrare le varie possibilità di tradurre una gerarchia di generalizzazione

Eliminazione delle entità figlie → E1,…, En vengono eliminate. Gli attributi di E1,…,En si inseriscono nellʼentità padre come attributi opzionali. Nellʼentità padre si inserisce un attributo tipo chex permette di distinguere le varie entità eliminazione dellʼentità padre → si può applicare sono nel caso di generalizzazione totale. Si inseriscono gli attributi di E in tutte le entità figlie. Ogni associazione a cui partecipava lʼentità padre, si sostituisce con n nuove associazioni, una per ogni entità figlia sostituzione della generalizzazione con associazioni → le entità non si modificano. Nelle associazioni la gerarchia si sostituisce con n associazioni uno a uno, ognuna delle quali lega lʼentità padre con una diversa entità figlia.

spiegare le operazioni che si compiono durante la ristrutturazione

Nella fase di ristrutturazione viene generato uno schema ER semplificato, quindi ristrutturato, ma equivalente a quello di partenza, al fine di semplificarne la traduzione successiva. si elimina dallo schema ER tutti quei costrutti non direttamente rappresentabili nel modello relazionale e vi sono ulteriori ristrutturazioni che tengano conto di aspetti prestazionali identificati dallʼanalisi del carico del lavoro. Laristrutturazione non è sempre univoca, poiché dipende dal carico del lavoro e da considerazioni relative alla conseguente realizzazione della base di dati che stiamo progettando. Lo schema ER ristrutturato tiene conto di aspetti realizzativi, per cui non è più uno “schema concettualeˮ. Si elimina dallo schema ER tutti i costrutti non direttamente rappresentabili nel modello relazionale, quindi gli attributi composti e multi-valore e le gerarchie di generalizzazione. Di eseguono ulteriori ristrutturazioni dello schema che tengano conto di aspetti prestazionali, identificati dallʼanalisi del carico del lavoro.

16. Spiegare cosa sono le viste e dare il comando di definizione in SQL illustrandone tutte le clausole e dare il comando sql per crearlo:

Una vista è una relazione virtuale. Il contenuto (tuple) è definito mediante un'interrogazione SQL sulla base dei dati (il contenuto della vista dipende dal contenuto delle altre relazioni di base presenti nella base di dati) Il contenuto non è memorizzato fisicamente nella base di dati, ma è ricalcolato tutte le volte che si usa la vista eseguendo l'interrogazione che la definisce. Una vista è un oggetto della base di dati, e può essere usata a quasi tutti gli effetti come una relazione di base. Il meccanismo delle viste è utile per: (1) semplificare l'accesso ai dati; (2) fornire indipendenza logica; (3) garantire la protezione dei dati CREATE VIEW <nome vista> [(<lista nomi colonne>] AS <interrogazione> [WITH [{LOCAL | CASCADED}] CHECK OPTION]; DROP VIEW nome\_vista dove: <nome vista> è il nome della vista <interrogazione> è l'interrogazione di definizione della vista. Le colonne della vista corrispondono in numero e dominio alle colonne specificate nella clausola di proiezione di tale interrogazione <lista nomi colonne> è una lista di nomi da assegnare alle colonne della

vista: La specifica non è obbligatoria, tranne nel caso in cui l'interrogazione contenga nella clausola di proiezione colonne virtuali cui non è assegnato un nome.

11.Facendo riferimento al modello ER spiegare cosa sono i vincoli di cardinalità massima e minima per le associazioni dando la definizione di associazione uno a molti, molti a molti ecc

): I vincoli di cardinalità per le associazioni esprimono vincoli di

partecipazione utili per modellare adeguatamente il dominio applicativo target. Vi è la cardinalità minima, ovvero il numero di istanze di unʼassociazione a cui unʼistanza delle entità coinvolte nellʼassociazione può partecipare; e la cardinalità massima, ovvero il numero massimo di istanze di unʼassociazione a cui una istanza delle entità coinvolte nellʼassociazione può partecipare. Associazione uno a molti → unʼassociazione binaria tra due entità E1 ed E2 si dice uno a molti se c\_max di E1 rispetto ad A è n e c\_max di E2 rispetto ad A è 1. c\_max di E1 rispetto ad A è 1 e c\_max di E2 rispetto ad A è n esempio azienda impiegato Associazione molti a molti → unʼassociazione binaria tra due entità E1 ed E2 si dice molti a molti se c\_max di E1 ed E2 rispetto ad A è n. esempio studente e corso

Associazioni uno a uno → unʼassociazione binaria tra due entità E1 ed E2 si dice uno a uno se c\_max di E1 ed E2 rispetto ad A è 1. esempio persona e passaporto Illustrare le varie tipologie di associazioni binarie che possono esistere in uno schema ER e dare un esempio per ognuna di esse (guardare domanda precedente)

spiegare le tipologie di vincoli che in SQL possono essere specificati ed illustrare la sintassi SQL per la loro specifica:

i vincoli di integrità sono verificati dopo ogni istruzione SQL che opera sulla tabella soggetta al vincolo: inserimento di nuovi dati e modifica del valore di colonne soggette al vincolo. Se il vincolo è violato, lʼistruzione SQL che ha causato la violazione genera un errore di esecuzione. In SQL si possono specificare diversi tipi di vincoli di

integrità: (NOT NULL) → specifica che un attributo non può essere nullo chiavi (PRIMARY KEY e UNIQUE) → primary key serve per la specifica della chiave primaria: impone che per ogni tupla i valori degli attributi

18. Spiegare cos’è una gerarchia di generalizzazione nel modello ER ed illustrarne le varie tipologie:

Unʼentità è una generalizzazione delle entità E1, En se ogni istanza delle entità E1, …, En è anche unʼistanza di E: E: entità padreE1, …, En entità figlie Ogni istanza delle entità figlie è anche istanza dellʼentità padre. Tutte le proprietà dellʼentità padre (attributi, identificatori ed associazioni) vengono ereditate dalle entità figlie. Vincoli impliciti: Se unʼentità E1 è definita come generalizzazione di una entità E2: lʼinsieme delle istanze E2 di deve essere contenuto in quello delle istanze di E1; ogni attributo di E1 è anche un attributo di E2; ad ogni associazione cui partecipa E1 partecipa anche E2. Entità generalizzazione di E1, …, En:

totale: ogni istanza di è istanza di almeno unʼentità es: Cliente - Standard, VIP, se standard e VIP sono le uniche tipologie di clienti previste dalla videoteca parziale: esiste almeno unʼistanza di che non è istanza di alcuna entità es: Cliente - Standard, VIP, se esistono clienti che possono effettuare un noleggio senza registrarsi al programma di fidelizzazione

La generalizzazione può essere inoltre: esclusiva: ogni istanza di è istanza di al più unʼentità es: Cliente - Standard, VIP condivisa: esiste almeno unʼistanza di che è istanza di più di unʼentità es: Film FilmAnimazione, FilmEssa Queste due classificazioni sono ortogonali. Le generalizzazioni possono quindi essere di 4 tipi: totali esclusive, totali condivise e parziali esclusive, parziali condivise

specificati come PK siano non nulli e diversi da quelli di ogni altra tupla.

Unique serve per specificare chiavi alternative e garantisce che non esistano due tuple che condividono gli stessi valori non nulli per gli attributi oggetto del

vincolo UNIQUE (le colonne UNIQUE possono contenere valori nulli). In una tabella è possibile specificare più chiavi UNIQUE ma una sola PRIMARY KEY (FOREIGN KEY) → la specifica di chiavi esterne avviene mediante la clausola (opzionale) FOREIGN KEY del comando CREATE TABLE. Se gli attributi riferiti hanno lo stesso nome di quelli referenti, non è obbligatorio specificarli. Lʼattributo/i riferiti deve/devono essere soggetto/i al vincolo unique o primary key. FOREIGN KEY (<lista colonne referenti>) REFERENCES <nome relazione>[(<lista colonne riferite>)] [ON DELETE {NO ACTION | CASCADE | SET NULL | SET DEFAULT}] [ON UPDATE {NO ACTION|CASCADE | SET NULL | SET DEFAULT} vincoli CHECK \* Nel comando CREATE TABLE è possibile definire: Vincoli CHECK su colonna e su relazione. Alla specifica della colonna viene affiancata la parola chiave CHECK seguita da una condizione, cioè un predicato o una combinazione booleana di predicati. Tale condizione può anche contenere sotto-interrogazioni che fanno riferimento ad altre relazioni. Alla definizione di una relazione viene aggiunta la parola chiave CHECK seguita da un predicato

o una combinazione booleana di predicati. La condizione può contenere sotto-interrogazioni che fanno riferimento ad altre tabelle.

4.Illustrare le caratteristiche più importanti del linguaggio SQL

è un linguaggio per la definizione e la manipolazione dei dati supportato dalla totalità dei DBMS relazionali: Structured Query Language. SQL è un linguaggio dichiarativo: descrive cosa fare e non come fare, si pone ad un livello di astrazione superiore rispetto ai linguaggi di programmazione tradizionali, è basato sullʼalgebra relazionale (per quanto riguarda le operazioni di query). SQL è un linguaggio set-oriented: gli operatori operano su relazioni; il risultato è sempre una relazione. Il linguaggio è utilizzabile in due modalità: interattiva; compilata; un linguaggio ospite(host) - > contiene le istruzioniSQL. Eʼ diventato standard ufficiale nel 1986 (SQL- 1986). Ultimo standard: SQL:2016. SQL comprende istruzioni per la definizione, lʼinterrogazione e lʼaggiornamento dei dati: è sia DDL che DML. Tutte le

implementazioni di SQL prevedono inoltre comandi SDL (non standardizzati)

spiegare componenti principali del modello ER:

modello Entità-Relazione è uno dei modelli più utilizzati nellʼambito della progettazione concettuale ed è dotato di una rappresentazione grafica: diagramma ER. I costrutti di base

sono: le ENTITÀ, ovvero una collezione di oggetti della realtà che vogliamo modellare e che possiedono caratteristiche comuni e con esistenza “autonomaˮ. Vi sono le istanze di entità,

ovvero specifici oggetti appartenenti ad una certa entità; le ASSOCIAZIONI o relazioni, ovvero il legame che vi è tra le entità. Le istanze di associazione sono combinazioni di istanze delle entità che prendono parte ad una associazione. Lʼinsieme delle istanze di unʼassociazione è un sottoinsieme del prodotto Cartesiano degli insiemi delle istanze delle entità che partecipano allʼassociazione; quindi, essa può contenere duplicati e ciò ha conseguenze sulle scelte progettuali. Il grado di unʼassociazione è il numero di entità che partecipano ad unʼassociazione e vi sono: quella unario, che modella una associazione

ricorsiva, quella binaria e quella n-aria.gli ATTRIBUTI, ovvero delle proprietà elementari possedute da unʼentità o da unʼassociazione di interesse nel dominio applicativo target, anche le associazioni possono avere attributi. Un attributo può essere anche composto, ovvero che raggruppa attributi di entità o associazione che presentano affinità nel loro uso o significato e possiede sotto- attributi

Spiegare cosʼè un’estensione procedurale di SQL ed illustrare le tipologie di istruzioni che deve fornire:

per poter integrare SQL ad un linguaggio di programmazione usare le estensioni procedurali di SQL. In questo modo SQL viene esteso con gli usuali costrutti dei linguaggi di programmazione. Esso può essere usato allʼinterno del DBMS per definire delle funzioni o delle procedure richiamabili da qualunque applicazione spiegare come si eliminano gli attributi composti e multi-valore da uno schema ER: per eliminare un attributo composto A da unʼentità E vi sono due soluzion la prima, ovvero attraverso lʼeliminazione dei sotto-attributi di A e lʼattributo composto che diventa un attributo semplice, è compito

dell'applicazione garantire che il nuovo attributo contenga valori coerenti con la semantica dell'attributo composto ristrutturato ed è adatta se è sufficiente lʼaccesso allʼinformazione complessiva; la seconda, ovvero con la considerazione di tutti i sotto-attributi di A come attributi di E e la ridefinizione del dominio dellʼattributo, si perde la relazione tra i sotto-attributi ed è adatta se è necessario accedere separatamente a ciascun attributo. Eventuali vincoli di cardinalità per lʼattributo composto vengono associati a ciascuno dei nuovi attributi generati tramite la ristrutturazione. Se le componenti dellʼattributo composto sono a loro volta attributi composti, si riapplica la procedura. Per eliminare degli attributi multi-valore, si definisce una nuova entità, collegata allʼentità di partenza tramite opportuna associazione e lʼattributo multi-valore è rappresentato mediante un attributo mono-valore che identifica lʼentità. Vi sono dei vincoli di cardinalità rispetto alla nuova associazione: per lʼentità che conteneva prima della ristrutturazione lʼattributo multi-valore, il vincolo coincide con il vincolo di cardinalità dellʼattributo multi-valore, invece per la nuova entità, dipende dal dominio.

spiegare come operano le metodologie top down, bottom up e mista nella progettazione di una base di dati

Top down \* Il processo di progettazione comincia da concetti molto astratti e generali, nel corso del progetto i singoli concetti sono approfonditi e arricchiti con i dettagli

Bottom up \* Consiste nel descrivere le componenti del sistema

partendo da uno schema concettuale dettagliato, dal basso (particolare) verso lʼalto (generale)

Misto \* Si individuano i concetti principali e si realizza uno schema scheletro, poi si raffina e si espande e infine si integra per arrivare allo schema finale

Spiegare cosa si intende per modello dei dati ed illustare i concetti principali che deve rappresentare:

una delle caratteristiche del DBMS è lʼefficienza, dovuta allo schema, o schema logico, che descrive il contenuto della base di dati con un formalismo ad alto livello, evitando lʼimplementazione fisica. Questo schema viene detto modello dati. È un insieme di concetti usato per la rappresentazione dei dati di nostro interesse e ha come componenti un insieme di strutture dati e i linguaggi per specificare i dati tramite strutture del modello, aggiornare le strutture, specificare i vincoli delle strutture e manipolare e ricercare i dati.

spiegare a cosa servono le clausole ON UPDATE e ON DELETE del comando di CREATE TABLE,

dare una definizione di ognuna delle opzioni possibili in tale clausole:

la clausola opzionale ON DELETE permette di specificare le azioni da eseguire quando vengono cancellate delle tuple nella tabella riferita. Analogamente, la clausola opzionale. ON UPDATE permette di specificare come gestire le modifiche dei valori della chiave primaria nella tabella riferita. Per entrambe le clausole, le opzioni disponibili sono:

NO ACTION : la cancellazione / modifica della tupla nella tabella viene eseguita solo se non ci sono riferimenti a essa nella tabella referente, altrimenti il DBMS rifiuta

l'operazione e segnala un errore

CASCADE : la cancellazione / modifica della tupla nella tabella riferita implica anche la cancellazione / modifica di tutte le tuple nella tabella referente che fanno riferimento a

essa.

SET NULL : la tupla nella tabella riferita viene cancellata / modificata e i valori della chiave esterna che facevano riferimento a essa vengono posti a NULL (se ammesso)

SET DEFAULT : la tupla nella tabella riferita viene cancellata / modificata, e i valori della chiave esterna che facevano riferimento a essa vengono posti al valore di

default (se specificato nella creazione della tabella)

L'opzione di default è NO ACTION . Osservazione: queste opzioni sono specificate nella creazione della tabella referente, anche se riguardano operazioni effettuate sulla tabella riferita. In questo modo, se ci sono più tabelle referenti per una stessa tabella riferita, ciascuna di esse può specificare comportamenti diversi, in base alle esigenze del dominio applicativo.