

Prova Compitino

- Lo schema di base dati sotto riportato, è usato per gestire le informazioni per la gestione della conservazione ed invecchiamento di vini in bottiglia.
- I vini sono raggruppati in lotti ciascuno dei quali può essere composto da un numero variabile di bottiglie di vino (Lotto.bottiglie), tutte della stessa annata e tipologia (Lotto.annata e Lotto.vino).
- Ogni tipologia di vino si distingue in base alla percentuale (Composizione.percentuale) con cui vengono miscelati un insieme predefinito di vitigni (Composizione.vitigno).
- Esiste un vincolo di integrità referenziale tra gli attributi e le relazioni che hanno lo stesso nome (per esempio, l'attributo vino in Lotto e la relazione Vino)

Lotto(codice, bottiglie, vino, annata)

Vino(tipo, nome)

Vitigno(codice, nome)

Composizione(vino, vitigno, percentuale)

Prova Compitino

Specificare, nel linguaggio SQL, le seguenti interrogazioni:

1. La tipologia (o le tipologie) di vino per cui si ha il maggior numero totale di bottiglie (anche su lotti diversi), indipendentemente dall'annata.
2. Il nome del vino (o dei vini) dell'annata più vecchia.
3. Il nome di ogni vitigno ed il numero di diverse tipologie di vino in cui è impiegato con percentuale maggiore del 20%.

Lotto(codice, bottiglie, vino, annata)

Vino(tipo, nome)

Vitigno(codice, nome)

Composizione(vino, vitigno, percentuale)

Prova Compitino II

- Lo schema di base dati sotto riportato, è usato per gestire le informazioni sulla carriera degli studenti di un corso di laurea.
- Esiste un vincolo di integrità referenziale tra gli attributi e le relazioni che hanno lo stesso nome

Studente(matricola, nome, cognome)

Iscrizione(studente, anno)

Esame(codice, nome)

EsamiSostenuti(studente, esame, voto, giorno, mese, anno)

Prova Compitino II

Specificare, nel linguaggio SQL, le seguenti interrogazioni:

1. Matricola, nome e cognome dello studente che ha sostenuto il maggior numero di esami in un anno
2. Codice dell'esame per il quale risulta minima la media dei voti attribuiti agli studenti
3. Matricola degli studenti che hanno dato più di un esame nello stesso giorno

Studente(matricola, nome, cognome)

Iscrizione(studente, anno)

Esame(codice, nome)

EsamiSostenuti(studente, esame, voto, giorno, mese, anno)

Prova Compitino III

- Lo schema di base dati sotto riportato, è usato per gestire le informazioni sui libri presenti in una biblioteca.
- Esiste un vincolo di integrità referenziale tra gli attributi e le relazioni che hanno lo stesso nome (per esempio, l'attributo libro in AutoriLibri e la relazione Libro)

Scrittore(cf, nome, cognome)
AutoriLibri(libro, scrittore)

Libro(isbn, titolo, editore)
Editore(codice, nome)

Prova Compitino III

Specificare, nel linguaggio SQL, le seguenti interrogazioni:

1. Il nome di ciascun editore ed il numero di libri che ha edito (per gli editori che non hanno edito libri deve comparire 0)
2. Codice isbn e titolo del libro con il massimo numero di autori
3. Per ciascuno scrittore che ha scritto almeno un libro, il codice fiscale ed il numero di editori per cui ha scritto dei libri.

Scrittore(cf, nome, cognome)
AutoriLibri(libro, scrittore)

Libro(isbn, titolo, editore)
Editore(codice, nome)

Prova Compitino IV

Di seguito è mostrato il modello relazionale di una base dati che organizza le informazioni di interesse per un'azienda. In particolare, nell'azienda vengono svolti alcuni progetti cui partecipano gli impiegati dell'azienda stessa. Alcuni impiegati svolgono il ruolo di supervisore per uno o più progetti.

Progetto (codice, nome, budget)

Impiegato (matricola, nome, cognome, stipendio)

Partecipazione (impiegato, progetto)

Supervisione (impiegato, progetto)

Vincoli di integrità referenziale:

attributo impiegato nella relazione Partecipazione e la relazione Impiegato

attributo impiegato nella relazione Supervisione e la relazione Impiegato

attributo progetto nella relazione Partecipazione e la relazione Progetto

attributo progetto nella relazione Supervisione e la relazione Progetto

Specificare in linguaggio SQL le interrogazioni atte ad estrarre i seguenti dati:

La matricola dell'impiegato che partecipa al maggior numero di progetti

Per ogni supervisore, il nome e cognome ed il numero di progetti che supervisiona

Per ogni progetto, lo stipendio medio (avg) degli impiegati che ci partecipano

Prova Compitino V

Lo schema sotto riportato rappresenta le informazioni di interesse per una società che vende prodotti ad utenti registrati. I dati relativi agli utenti sono memorizzati nella tabella User che include un numero di identificazione (uin), nome, cognome ed indirizzo.

I dati sui prodotti sono memorizzati nella tabella Product che include un numero di identificazione (pin), nome e costo unitario del prodotto.

Gli ordini effettuati dagli utenti sono memorizzati nella tabella Order. Ogni ordine può includere diversi prodotti e questo richiede che la tabella Order abbia come chiave il numero di identificazione dell'ordine (oin) ed il numero di identificazione del prodotto (product). Ogni riga riporta anche il numero di elementi richiesti per ciascun prodotto.

Lo stato di ciascun ordine è memorizzato nella tabella OrderStatus che include il numero d'ordine, il numero di identificazione dell'utente che ha effettuato l'ordine, la data di effettuazione dell'ordine e lo status dell'ordine:

- status=0 significa "Unavailability of some products"
- status=1 significa "All products available"
- status=2 significa "Products sent"
- status=3 significa "Delivery completed"

Il significato di ciascuno stato è descritto nella tabella Status che include il numero di identificazione di stato (0, 1, 2 and 3) insieme alla sua descrizione.

Quando un attributo ed una tabella hanno lo stesso nome si deve assumere esistere un vincolo di integrità referenziale tra l'attributo e la chiave primaria della tabella

User(uin, name, surname, address) **Product(pin, name, cost)**
Order(oin, product, number) **Status(sin, description)**
OrderStatus(oin, user, dateOfIssue, status)

Prova Compitino V

Assumendo che la funzione Elapsed(dateOfIssue) restituisca il numero di giorni trascorsi dalla dateOfIssue, scrivere in SQL le seguenti interrogazioni:

1. Per ciascun possibile stato degli ordini, la descrizione ed il numero effettivo di ordini in quello stato
2. Tutti i dati sull'utente che da più tempo sta ancora aspettando la consegna dell'ordine
3. Il numero medio di pezzi ordinati nei diversi ordini

User(uin, name, surname, address) **Product(pin, name, cost)**
Order(oin, product, number) **Status(sin, description)**
OrderStatus(oin, user, dateOfIssue, status)

Prova compitino VI

- Lo schema sotto riportato rappresenta le informazioni di interesse per la gestione del sistema di prenotazione posti alle rappresentazioni teatrali
- Per gli utenti sono di interesse il cf, nome, cognome e recapito telefonico
- Uno spettacolo è caratterizzato da un codice ed ha una descrizione. Ciascuno spettacolo può essere programmato in una o più date. Ogni giorno è in programmazione al più un solo spettacolo
- Per ciascuna serata in programmazione si possono effettuare delle prenotazioni su posti numerati
- I posti sono numerati univocamente e raggruppati in categorie cui corrispondono diverse fasce di costo
- La categoria di appartenenza di ciascun posto è esplicitata nella relazione CategoriePosti

Utente(cf, nome, cognome, telefono) **Spettacolo**(codice, descrizione)
Programmazione(data, spettacolo) **CategoriePosti**(posto, categoria)
Prenotazione(data, posto, utente)
Categoria(codice, descrizione, costo)

Prova compitino VI

- Specificare le seguenti interrogazioni in linguaggio SQL:
 - Descrizione dello spettacolo in programmazione il 15-12-2006 e numero dei posti prenotati
 - I dati di ciascun utente ed il costo totale dei biglietti da lui prenotati, ordinando i risultati per costo decrescente
 - Per ciascuna categoria, il nome ed il numero medio di posti prenotati negli spettacoli successivi al 10-1-06

Utente(cf, nome, cognome, telefono) Spettacolo(codice, descrizione)
Programmazione(data, spettacolo) CategoriePosti(posto, categoria)
Prenotazione(data, posto, utente)
Categoria(codice, descrizione, costo)

Prova compitino VII

- Lo schema sotto riportato rappresenta i dati di interesse per monitorare le multe effettuate da vigili

Policeman(code, name, surname, address, areaCode)

Fine(code, policeman, car, date, infraction, streetName, areaCode, cost)

Infraction(code, description)

Car(plateNumber, type, owner)

Owner(code, name, surname, address, areaCode)

AreaCode(code, cityName, provinceName)

Prova compitino VII

- Codice, nome e cognome di ogni vigile e numero di multe fatte, ordinando i risultati per valori decrescenti del numero di multe
- Per ogni CAP, il codice ed il numero medio di multe fatte in un giorno
- Il CAP dove sono state fatte il maggior numero di multe, insieme a tale numero

Policeman(code, name, surname, address, areaCode)

Fine(code, policeman, car, date, infraction, streetName, areaCode, cost)

Infraction(code, description) **Car**(plateNumber, type, owner)

Owner(code, name, surname, address, areaCode)

AreaCode(code, cityName, provinceName)

Prova compitino VIII

- Stesso schema precedente.
 - Per ogni CAP, il numero di multe fatte a residenti nello stesso CAP ed a residenti fuori ordinando i risultati per CAP crescente
 - Codice della persona a cui sono state fatte più multe, il numero di multe fatte ed il costo totale di tali multe
 - Per ogni vigile il suo codice ed il numero medio di multe fatte al giorno negli ultimi 30 giorni (date>"23/10/2010")
 - Targa della vettura a cui sono state fatte più multe, il numero di multe fatte ed il costo totale di tali multe
 - Per ogni proprietario il suo codice ed il numero medio di multe ricevute al giorno negli ultimi 30 giorni (date>"23/10/2010")
 - Per ogni CAP, il costo totale delle multe fatte a residenti nello stesso CAP ed a residenti fuori ordinando i risultati per CAP crescente

Policeman(code, name, surname, address, areaCode)

Fine(code, policeman, car, date, infraction, streetName, areaCode, cost)

Infraction(code, description)

Car(plateNumber, type, owner)

Owner(code, name, surname, address, areaCode)

AreaCode(code, cityName, provinceName)

Prova compitino IX

- Lo schema sotto riportato rappresenta i dati di interesse per monitorare le visite effettuate ai pazienti di uno studio medico. **Paziente** contiene i dati dei pazienti con relativi recapiti; **Medico** contiene i dati relativi ai medici dello studio. I dati relativi alla prenotazione di una visita sono memorizzati in **PrenotazioneVisita**. Per ogni visita realmente fatta (il paziente che ha prenotato potrebbe poi non presentarsi o disdire) viene aggiunta una riga alla tabella **Visita**. **DettaglioVisita** contiene i dati sui controlli effettuati nella visita con una indicazione sull'esito (normale true/false) di ciascun controllo. Le prescrizioni fatte al termine di una visita sono memorizzate nella tabella **Prescrizione**, mentre la tabella **Farmaco** esplicita il nome di ciascun farmaco prescrivibile.
- Sussiste un vincolo di integrità referenziale tra gli attributi e le relazioni che hanno lo stesso nome (per esempio paziente in **Visita** e la chiave primaria di **Paziente**)

Paziente(cf, nome, cognome, indirizzo, città, telefono)

Medico(cf, nome, cognome, indirizzo, città, telefono)

PrenotazioneVisita(data, ora, paziente, medico)

Visita(codice, data, paziente, medico, costo)

DettaglioVisita(visita, controllo, inNorma, descrizione)

Prescrizione(visita, farmaco, posologia)

Controllo(codice, nome)

Farmaco(codice, nome)

Prova compitino IX

- Il numero di visite al termine delle quali non e' stato prescritto il farmaco F018
 - Il numero di visite in cui non e' stato effettuato il controllo C014
- Il medico che ha visitato il maggior numero di diversi pazienti
 - Il paziente che è stato visitato dal maggior numero di diversi medici
- Il numero medio di farmaci prescritti al termine delle visite
 - Il numero medio di controlli fatti in una visita

Paziente(cf, nome, cognome, indirizzo, citta, telefono)

Medico(cf, nome, cognome, indirizzo, citta, telefono)

PrenotazioneVisita(data, ora, paziente, medico)

Visita(codice, data, paziente, medico, costo)

DettaglioVisita(visita, controllo, inNorma, descrizione)

Prescrizione(visita, farmaco, posologia)

Controllo(codice, nome)

Farmaco(codice, nome)