Basi di Dati e Web

Esercitazione del 28 Marzo 2012

SQL

Esercizio 1 (3 Febbraio 2003)

Si considerino i seguenti schemi relazionali:

CLIENTE(Id, Cognome, Nome, NumNoleggi, Tipologia)
FILM(Titolo, Regista, Genere, Durata)
CASSETTA(Id, Titolo)
DVD(Id, Titolo)
NOLEGGIO(IdCassODvd, CodCliente, DataPrestito, DataRestituzione)

1. Estrarre in SQL tutti i film di cui sono stati noleggiati più volte i DVD che le videocassette.
2. Estrarre in SQL tutti i film le cui cassette sono state noleggiate più di venti volte nel mese di Gennaio ma non presenti in formato DVD.
3. * Determinare il cognome e il nome dei clienti che hanno noleggiato tutti i film (in formato cassetta o DVD).
4. * Determinare il nome e cognome dei clienti che hanno noleggiato DVD solo di genere giallo.

Soluzione

1.
```
CREATE VIEW DVD (Titolo, Numero) AS
            SELECT   Titolo, COUNT(*)
            FROM     DVD JOIN Noleggio ON Id = IdCassODvd
            GROUP BYTitolo

CREATE VIEW Cass (Titolo, Numero) AS
            SELECT   Titolo, COUNT(*)
            FROM     Cassetta JOIN Noleggio ON Id = IdCassODvd
            GROUP BYTitolo

SELECT   Titolo
FROM     DVD JOIN Cass ON DVD.Titolo = Cass.Titolo
WHERE    DVD.Numero > Cass.Numero
```

2.
```
SELECT   Titolo
FROM     Cassetta JOIN Noleggio ON Id = IdCassODvd
WHERE    DataPrestito BETWEEN 1/1/2012 AND 31/1/2012 AND Titolo NOT IN  (
             SELECT   Titolo
             FROM     DVD
             )

GROUP BYTitolo
HAVING   COUNT(*) > 20
```

3.

```
CREATE VIEW Supporto (Id, Titolo) AS
            SELECT  Id, Titolo
            FROM    Cassetta
            UNION
            SELECT  Id, Titolo
            FROM    DVD

SELECT  Cognome, Nome
FROM    Cliente AS C
WHERE   NOT EXISTS (
            SELECT  *
            FROM    Film AS F
            WHERE   NOT EXISTS (
                        SELECT  *
                        FROM    Noleggio JOIN Supporto ON IdCassODvd = Id
                        WHERE   Titolo = F.Titolo AND CodCliente = C.Id
                        )

            )
```

4.

```
SELECT  Cognome, Nome
FROM    Cliente AS C JOIN Noleggio AS N ON C.Id = CodCliente
                     JOIN DVD AS D ON IdCassODvd = D.Id
WHERE   NOT EXISTS (
            SELECT  *
            FROM    Noleggio AS N1 JOIN DVD AS D1 ON D1.Id = N1.IdCassODvd
                                   JOIN Film AS F1 ON D1.Titolo = F1.Titolo
            WHERE   C.Id = N1.CondCliente AND Genere ≠ 'Giallo'
            )
```

Esercizio 2

Si considerino i seguenti schemi relazionali:

PARTITA(Id, SquadreCasa, SquadraOspite, PuntiCasa, PuntiOspite)

1. Estrarre in SQL la classifica delle squadre, considerando che le vittorie valgono 3 punti e i pareggi 1 punto.

Soluzione

1.

```
CREATE VIEW PuntiPartita (Squadra, Punti) AS
            SELECT  SquadraCasa, 3
            FROM    Partita
            WHERE   PuntiCasa > PuntiOspite
            UNION ALL
            SELECT  SquadraOspite, 3
            FROM    Partita
            WHERE   PuntiCasa < PuntiOspite
            UNION ALL
            SELECT  SquadraCasa, 1
            FROM    Partita
            WHERE   PuntiCasa = PuntiOspite
            UNION ALL
            SELECT  SquadraOspite, 1
            FROM    Partita
            WHERE   PuntiCasa = PuntiOspite

CREATE VIEW PuntiTotali (Squadra, Punti) AS
            SELECT  Squadra, SUM(Punti)
            FROM    PuntiPartita
            GROUP BYSquadra

SELECT  Squadra
FROM    PuntiTotali
ORDER BYPunti DESC
```

Esercizio 3 (8 Settembre 2010)

Si considerino i seguenti schemi relazionali:

AEROPORTO(Id, Città, Nazione, NumPiste)
VOLO(NumeroVolo, GiornoSett, IdAeropPartenza, IdAeropArrivo, Compagnia, OraPartenza, OraArrivo, CodAereo)
AEREO(Codice, Tipo, NumPasseggeri)

1. Esprimere in SQL la query che estrae la coppia di città tra cui viene offerta la maggiore capacità complessiva di trasporto passeggeri. (Se si suppone che la capacità sia simmetrica, come si potrebbe rendere la query più efficiente?)
2. Estrarre in SQL le compagnie che servono la città di Bergamo con più di settanta voli alla settimana, usando aerei di un solo tipo per tutti i loro voli.

Soluzione

1.

```
CREATE VIEW Trasporto (Da, A, Passeggeri) AS
            SELECT   A1.Città, A2.Città, SUM(NumPasseggeri)
            FROM     Volo JOIN Aereo ON CodAereo = Codice
                          JOIN Aeroporto AS A1 ON IdAeropPartenza = A1.Id
                          JOIN Aeroporto AS A2 ON IdAeropArrivo = A2.Id
            GROUP BYA1.Città, A2.Città

CREATE VIEW Totali (CUno, CDue, Passeggeri) AS
            SELECT   T1.Da, T1.A, T1.Passeggeri+T2.Passeggeri
            FROM     Trasporto AS T1 JOIN Trasporto AS T2 ON T1.Da = T2.A AND T2.Da = T1.A
            WHERE    T1.Da < T1.A
            UNION
            SELECT   Da, A, Passeggeri
            FROM     Trasporto
            WHERE    Da NOT IN  (
                          SELECT   A
                          FROM     Trasporto
                          )


SELECT   DISTINCT CUno, CDue
FROM     Totali
WHERE    Passeggeri =  (
             SELECT   MAX(Passeggeri)
             FROM     Totali
             )
```

2.

```
CREATE VIEW Bg (Compagnia) AS
            SELECT   Compagnia
            FROM     Volo JOIN Aeroporto AS A1 ON IdAeropPartenza = A1.Id
                          JOIN Aeroporto AS A2 ON IdAeropArrivo = A2.Id
            WHERE    A1.Città = 'Bergamo' OR A2.Città = 'Bergamo'
            GROUP BYCompagnia
            HAVING   COUNT(*) ≥ 70

SELECT   Compagnia
FROM     Volo AS V JOIN Aereo AS A ON CodAereo = Codice
                   JOIN Bg ON V.Compagnia = Bg.Compagnia
WHERE    NOT EXISTS (
             SELECT   *
             FROM     Volo AS V1 JOIN Aereo AS A1 ON CodAereo = Codice
             WHERE    A1.Tipo ≠ A.Tipo AND V1.Compagnia = V.Compagnia
             )
```

Esercizio 4 (30 giugno 2010)

Si considerino i seguenti schemi relazionali:

PAGINA(Path, ContenutoHTML, Descrizione, TimestampUltimaModifica, IdProprietario)
ACCESSO(PathPagina, Tempo, IdUtente, PathPaginaProvenienza)
UTENTE(Id, Nome, Categoria)

1. Estrarre in SQL per ogni categoria di utenti il numero totale di accessi dall'esterno nel mese di giugno 2010 (si considerino accessi esterni quelli che presentano una pagina di provenienza esterna al sito).
2. Estrarre in SQL le pagine cui è stato fatto accesso entro 1 ora dopo la modifica della pagina.

Soluzione

1.
```
SELECT   Categoria,COUNT(*)
FROM     Accesso JOIN Utente ON IdUtente = Id
WHERE    Tempo BETWEEN 1/6/2012 AND 30/6/2012 AND PathPaginaProvenienza NOT IN  (
             SELECT   Path
             FROM     Pagina
             )

GROUP BYCategoria
```

2.
```
SELECT   DISTINCT Path
FROM     Pagina JOIN Accesso ON Path = Pagina
WHERE    Tempo.hour-TimestampUltimaModifica.hour < 1 AND Tempo.hour-TimestampUltimaModifica.hour > 0
```

Esercizio 5 (5 Giugno 2007)

Si considerino i seguenti schemi relazionali:

CONTO(Codice,NomeCorrentista,Saldo)
VERSAMENTO(IdVersamento, CodConto, Ammontare, Giorno, Mese, Anno)

1. Estrarre in SQL i correntisti che hanno avuto un totale mensile versato in almeno 2 mesi dell'anno 2006 superiore al doppio della media mensile, valutata su tutti i conti e su tutti i mesi di tutti gli anni.

Soluzione

1.

```
CREATE VIEW Versamenti (CC, Mese, Anno, Importo) AS
            SELECT   CodConto, Mese, Anno, SUM(Ammontare)
            FROM     Versamento
            GROUP BYCodConto, Mese, Anno

SELECT   Codice, NomeCorrentista
FROM     Versamenti JOIN Conto ON CC = Codice
WHERE    Anno = 2006 AND Importo ≥ ALL (
             SELECT   2∗AVG(Importo)
             FROM     Versamenti
             )

GROUP BYCodice, NomeCorrentista
HAVING   COUNT(*) ≥ 2
```