

# Compito di Basi di Dati e Sistemi Informativi

29 marzo 2007

## Esercizio 1:

Sia dato il seguente schema relazionale:

*Squadra*(*nomeSquadra*, *città*, *sponsor*, *allenatore*);

*Giocatore*(*numeroTessera*, *squadra*, *annoNascita*, *ruolo*);

*Partita*(*idPartita*, *giornata*, *nomeSqCasa*, *nomeSqTrasferta*, *goalCasa*, *goalTrasferta*).

Si assuma che ogni squadra sia identificata univocamente dal suo nome, ogni giocatore dal numero della sua tessera e ogni partita dal suo id.

Definire preliminarmente le chiavi primarie e le eventuali chiavi esterne delle relazioni date.

Successivamente, stabilire quali delle seguenti interrogazioni si possono esprimere in algebra relazionale.

In tutti i casi in cui risulta possibile, formulare opportune interrogazioni in algebra relazionale che permettano di determinare (senza usare l'operatore di divisione e usando solo se necessario le funzioni aggregate):

- (a) le squadre che in ogni partita giocata in casa hanno segnato almeno un goal;
- (b) le squadre che non hanno più di due giocatori disponibili per uno stesso ruolo;
- (c) le squadre che non hanno mai perso un derby (un derby è una partita tra due squadre della stessa città);
- (d) la giornata (le giornate se più di una) in cui sono stati segnati più goal, limitatamente alle giornate in cui c'è stata almeno una vittoria fuori casa e almeno un pareggio;
- (e) le squadre che hanno sconfitto un sottoinsieme proprio delle squadre sconfitte dall'Albinoleffe.

## Esercizio 2:

Con riferimento all'Esercizio 1, stabilire quali interrogazioni si possono esprimere nel linguaggio SQL (si faccia riferimento allo standard SQL-92). In tutti i casi in cui risulta possibile, formulare opportune interrogazioni in SQL che permettano di determinare quanto richiesto (usando solo se necessario le funzioni aggregate).

## Esercizio 3:

Si vuole progettare una base di dati di supporto alla gestione della programmazione cinematografica giornaliera di un certo insieme di comuni italiani (di tale insieme non fanno parte i capoluoghi di regione).

Per ogni comune, vogliamo memorizzare il nome, il numero di residenti, l'indirizzo del municipio e la regione di appartenenza. Di ogni regione vogliamo memorizzare il nome, il numero di abitanti e il capoluogo di regione. Comuni con lo stesso nome possono appartenere solo a regioni diverse. In ogni comune sono presenti dei cinema.

Per ogni cinema, vogliamo memorizzare il nome, l'indirizzo, il numero di telefono e il numero di sale disponibili. Cinema con lo stesso nome possono essere presenti solo in comuni diversi. Ogni cinema ha un certo numero di sale di proiezione (almeno una).

Per ogni sala, vogliamo mantenere informazioni sul nome della sala (sale con lo stesso nome possono appartenere solo a cinema diversi) ed il numero di posti disponibili. In ogni sala vengono proiettati dei film ad orari prestabiliti.

Per ogni film, vogliamo memorizzare il titolo, il nome e il cognome del regista e la durata. Ogni film viene proiettato in una o più sale. In una stessa sala possono essere proiettati anche film diversi, in orari diversi. Per ogni proiezione, vogliamo memorizzare l'orario d'inizio e la durata prevista della proiezione.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte.

Lo schema dovrà essere completato, qualora necessario, con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione.

Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione.

Se necessario, si introducano appropriate regole aziendali (regole di derivazione e vincoli di integrità).

#### **Esercizio 4:**

Si descrivano brevemente i vincoli intrarelazionali e interrelazionali supportati dal linguaggio SQL. Con riferimento al vincolo di integrità referenziale, si illustrino le modalità di definizione di tale vincolo in SQL e le politiche di reazione alle possibili violazioni.