

# Compito di Basi di dati

18 luglio 2023

## Esercizio 1:

Sia dato il seguente schema di una base di dati relazionale che registra informazioni relative a elezioni politiche mediante le seguenti tre tabelle:

*Politici*(*CodiceFiscale*, *Nome*, *Cognome*, *AnnoNascita*);

*Partiti*(*NomePartito*, *Segretario*, *Sede*);

*Eletti*(*Politico*, *Partito*, *AnnoElezione*).

Si assuma che ogni politico sia identificato univocamente dal suo codice fiscale e sia caratterizzato da nome, cognome e anno di nascita.

Si assuma, inoltre, che ogni partito sia identificato univocamente dal nome e sia caratterizzato dal politico che attualmente ricopre la carica di segretario e dall'indirizzo della sede nazionale.

Infine, di ogni elezione si registrino i politici eletti e i partiti con cui si sono presentati, più l'anno in cui si è svolta l'elezione. Si assuma che un politico possa essere eletto più volte, non necessariamente con lo stesso partito, e che ogni anno si sia svolta al più una tornata elettorale.

Definire preliminarmente le chiavi primarie, le eventuali altre chiavi candidate e, se ve ne sono, le chiavi esterne delle relazioni date. Successivamente, formulare opportune interrogazioni in SQL che permettano di determinare (senza usare l'operatore CONTAINS e usando solo se e quando necessario le funzioni aggregate):

- (a) i partiti che nella tornata elettorale del 2020 hanno eletto non più di 2 persone nate dopo il 1990;
- (b) per ogni elezione, il partito (i partiti se più di uno) che ha eletto il maggior numero di candidati.

## Esercizio 2:

Sia dato il seguente insieme di requisiti relativi ad una base di dati per la gestione di un insieme di officine.

- Ogni officina sia identificata univocamente dal nome e sia caratterizzata da un indirizzo, un recapito telefonico, un indirizzo email, un direttore e il numero di dipendenti. Si assuma che ogni officina abbia uno ed un solo direttore e almeno un dipendente e che un direttore diriga una ed una sola officina.
- Dipendenti e direttori siano identificati univocamente dal loro codice fiscale e siano caratterizzati dal nome, dal cognome, dall'indirizzo di residenza, da uno o più recapiti telefonici e dal numero di anni di servizio presso l'officina. Si tenga traccia anche dell'età dei direttori. Si assuma che un direttore possa essere un dipendente dell'officina che dirige, ma non sia necessariamente tale (ossia, in alcuni casi, il direttore possa non essere un dipendente dell'officina che dirige).
- Ogni riparazione venga effettuata da una ed una sola officina e riguardi uno ed un solo veicolo. Di ogni riparazione vengano registrati il codice (univoco nell'ambito dell'officina), l'ora e la data di accettazione del veicolo e, nel caso di riparazione terminata, l'ora e la data di riconsegna del veicolo.
- Dei veicoli vengano registrati modello, tipo, targa, anno di immatricolazione e proprietario. Ogni veicolo abbia uno ed un solo proprietario. Dei proprietari di veicoli vengano registrati il codice fiscale, il nome e il cognome, l'indirizzo di residenza e uno o più recapiti telefonici.

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 con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione. Si definiscano anche eventuali regole di gestione (regole di derivazione e vincoli di integrità) necessarie per codificare alcuni dei requisiti attesi del sistema.

### Esercizio 3:

Si immagini di voler progettare una base di dati che memorizza informazioni relative ai **reperti** archeologici esposti e conservati nelle **sale** di un museo.

Ogni **reperto** sia identificato da un **codice** alfanumerico di lunghezza 5. Di ogni reperto si vogliono memorizzare la data in cui è stato **scoperto** e la data in cui è stato **consegnato** al museo, con l'accortezza che la seconda sia sempre successiva alla prima. Di ogni reperto si vuole sapere la **collocazione**, ovvero la sala in cui è esposto (questo attributo non può essere nullo).

Ogni **sala** è identificata da un **codice** di al più tre cifre ed è caratterizzata dal **nome** (che può essere nullo), dall'**apertura** o meno al pubblico e dalla **capacità espositiva**, ovvero il numero massimo di manufatti che la sala può ospitare (tale valore viene espresso mediante un numero intero maggiore di zero). Esiste una sala speciale, chiamata magazzino, che ospita i reperti non esposti in altre sale, la cui capacità espositiva si assume essere illimitata (per semplicità, un intero molto grande).

Si scriva il codice SQL per definire le due tabelle **reperto** e **sala**, definendo gli opportuni vincoli di integrità, sia interrelazionali che dei singoli attributi, facendo ragionevoli assunzioni dove non espressamente specificato. Un esempio valido delle tabelle popolate è il seguente:

Table 1: **REPERTO**

codice	data_scoperta	data_consegna	collocazione
aaaa1	01/06/2022	01/07/2022	100
abb23	28/01/2022	31/01/2022	111
c12a1	13/10/2022	17/11/2022	111

Table 2: **SALA**

codice	nome	aperta	capacita_expo
100	Magazzino	False	999999
111	Paleolitico	True	15

Si ipotizzi di voler inserire un nuovo reperto (*bb123*) e, contestualmente, si realizzi una nuova sala espositiva (*Antica Grecia*), non già presente nella basi di dati, ove inserire tale reperto. Si scriva il codice SQL per lo svolgimento di tali operazioni, tenendo conto del loro ordine di esecuzione.

Successivamente, si scriva il codice SQL per cancellare dalla base di dati il reperto *c12a1*.

Infine, si implementi un trigger SQL che gestisca il caso in cui un reperto venga assegnato ad una stanza che ha già raggiunto la massima capacità espositiva. In tale situazione, il reperto dovrà essere collocato automaticamente nel magazzino (in alternativa, l'operazione dovrà essere annullata).

### Esercizio 4:

Si descrivano brevemente le proprietà di atomicità e di isolamento delle transazioni e si spieghi come esse vengono garantite dai DBMS. Successivamente, si risponda alle seguenti domande, giustificando brevemente le risposte.

- Sia *t* una transazione operante su un dato insieme di tabelle. Può *t* eseguire un commit degli aggiornamenti effettuati su un sottoinsieme proprio delle tabelle, senza eseguire contemporaneamente un commit degli aggiornamenti effettuati sulle tabelle rimanenti?
- L'annidamento delle transazioni è consentito o meno?