

Nome:	Cognome:	Matricola:
--------------	-----------------	-------------------

Esercizio 1

Si consideri la seguente tabella contenente le informazioni che compaiono nel catalogo di un museo: il museo è diviso su più edifici aventi indirizzi diversi, ciascun edificio ha più piani e varie stanze su ogni piano; le stanze possono avere nomi uguali in edifici diversi, ma tutti diversi nello stesso edificio indipendentemente dal piano. Ogni oggetto esposto in un edificio ha un codice e una descrizione diversa, mentre il nome di oggetti diversi può essere uguale. Ogni Sorvegliante ha assegnate per la sorveglianza più stanze dello stesso edificio e sullo stesso piano.

Catalogo(CodOggetto, DescrizioneOggetto, NomeOggetto, NomeEdificio, Stanza, Piano, IndirizzoEdificio, Sorvegliante)

a) Individuare la chiave e tutte le dipendenze funzionali non banali

Supponendo che ogni oggetto abbia un codice diverso, dovunque sia esposto
 CodOggetto → DescrizioneOggetto, IndirizzoEdificio, Stanza
 IndirizzoEdificio, Stanza → Piano, Sorvegliante
 DescrizioneOggetto → NomeOggetto

Chiave: CodiceOggetto oppure DescrizioneOggetto

b) Verificare se Catalogo è in BCNF e, eventualmente, portarla in BCNF.

Non BCNF

Descrizione(CodOggetto, DescrizioneOggetto, IndirizzoEdificio, Stanza)
 LogisticaMuseo(IndirizzoEdificio, Stanza, Piano, Sorvegliante)
 IndiceNomi(DescrizioneOggetto, NomeOggetto)

Esercizio 2

Dare la definizione di chiusura transitiva di un insieme di dipendenze.

La chiusura di un insieme di dipendenze è $F^+ = \{X \rightarrow Y \mid F \Rightarrow X \rightarrow Y\}$

Esercizio 3

Perché i sistemi NoSQL non sono basati sulle proprietà ACID? Su quale paradigma sono basati?

I sistemi noSQL sono basati sul paradigma Base, ovvero Basically Available, Soft state, Eventual consistency, invece che godere delle proprietà di Atomicità, Coerenza, Isolamento e Durabilità. La coerenza dei dati in un sistema sempre disponibile e altamente distribuito non è garantibile.

Esercizio 4

Considerare la seguente base di dati:

CLIENTI (Codice, Nome, Cognome, Indirizzo, Città)

NOLEGGI (CodCliente, TargaAuto, GiornoPrelievo, MesePrelievo, AnnoPrelievo, GiornoRestituzione, MeseRestituzione, AnnoRestituzione, LuogoNoleggio)

AUTOVETTURE (Targa, Modello, Colore, AnnoImmatricolazione, CostoGiornaliero)
INCIDENTI (Targa, GiornInc, MeseInc, AnnoInc, Targa2, CostoRip, GiorniFermo)

a) Scrivere una espressione in algebra relazionale che elenchi Nome e Cognome dei clienti che, con un'auto noleggiata nel 2013, hanno avuto incidenti con un costo superiore a 1000 euro. Il costo di un incidente si calcola come il costo della riparazione più il costo giornaliero della macchina per il numero di giorni di fermo.

$$\Pi_{N,C} (\Pi_{N,C,C}(\text{Clienti}) \triangleright \triangleleft_{(CC=C)} \Pi_C (\sigma_F \Pi_{C,GP,MP,AP,GR,MR,AP,GI,MI,AI} (\sigma_{AP='2013 \text{ Noleggi}} \triangleright \triangleleft_{(TA=T)} \Pi_{T,GI,MI,AI} (\sigma_{(\text{CostoGiornaliero} * \text{GiorniFermo} + \text{CostoRip} > 1000)} (\text{Autovetture} \triangleright \triangleleft \text{Incidenti})))$$

$$F = ((AP=AI) \wedge (((MP < MI) \wedge ((MI < MR) \vee (MR = \text{NULL}))) \vee ((MP = MI) \wedge ((GP \leq GI) \wedge ((GI < GR) \vee (GR = \text{NULL}))) \vee ((MI = MR) \wedge ((GI < GR) \vee (GR = \text{NULL})))) \vee ((AP < AI) \wedge ((AI < AR) \vee (AR = \text{NULL}))) \wedge (((MP < MI) \wedge ((MI < MR) \vee (MR = \text{NULL}))) \vee ((MP = MI) \wedge ((GP \leq GI) \wedge ((GI < GR) \vee (GR = \text{NULL}))) \vee ((MI = MR) \wedge ((GI < GR) \vee (GR = \text{NULL}))))$$

b) Formulare l'interrogazione precedente nel calcolo dei domini.

{Cognome:c, Nome:n | Clienti(Codice:c, Cognome:c, Nome:n, Città:ct,...) ∧ Noleggi(CodCliente:c, TargaAuto: t, GiornoPrelievo:gp, MesePrelievo:mp, AnnoPrelievo:ap, Giornorestituzione:gr, MeseRestituzione:mr, AnnoRestituzione:ar, LuogoNoleggio:ct) ∧ Autovetture(Targa:t, CostoGiornaliero:gc,...) ∧ Incidenti(Targa:t, GiornInc:gi, MeseInc:mi, AnnoInc:ai, CostoRip:cr, GiorniFermo:gf,...) ∧ F ∧ (cg*gf+cr>1000) ∧ ap='2013' }

Esercizio 5

Verificare se i seguenti schedule sono tra loro view-equivalent o conflict-equivalent fornendo lo schedule seriale equivalente, se esiste.

$w2(x), r1(x), w1(x), r3(y), w1(y), r2(y), r3(z), r1(z), w3(z)$

$w2(x), r1(x), r2(y), w1(x), w1(y), r3(y), r3(z), r1(z), w3(z)$

Non son view-equivalenti perché nel primo $r2(y)$ legge quello che ha scritto $w1(y)$, non sono neanche conflict-equivalenti perché i grafi dei conflitti sono diversi.