

COGNOME

NOME

MATRICOLA

Basi di Dati – 2 luglio 2021

Esercizio 1. (NORMALIZZAZIONE)

Considerare la relazione seguente:

ID	CM	Mittente	CDes	Destinazione	CCD	Categoria	CT	Tipo	NP	CDim	Dimens.
11	M1	M. Rossi	D1	Neri Via ...	A	privato	T1	Normale	1	P	Piccolo
11	M1	M. Rossi	D1	Neri Via ...	A	privato	T1	Normale	2	M	Medio
12	M1	M. Rossi	D2	Bisi Via ...	B	ufficio	T2	Urgente	1	M	Medio
13	M2	G. Bruni	D2	Bisi Via ...	B	ufficio	T2	Urgente	1	G	Grande
13	M2	G. Bruni	D2	Bisi Via ...	B	ufficio	T2	Urgente	2	G	Grande
14	M3	S. Verdi	D3	Cini P.zza ...	B	ufficio	T1	Normale	1	G	Grande

La relazione mostra (in forma non normalizzata) i dati relativi ad un insieme di spedizioni, secondo le seguenti specifiche:

- ogni spedizione ha un codice (ID), un mittente, una destinazione e un tipo
- ogni mittente ha un codice (CM) e un nome (attributo Mittente)
- ogni destinazione ha un codice (CDes), una descrizione (attributo Destinazione) e una categoria, con codice (CCD) e descrizione (Categoria)
- ogni tipo di spedizione ha un codice (CT) e una descrizione (attributo Tipo)
- ogni spedizione comprende uno o più pacchi
- ogni pacco ha un numero progressivo (NP) che lo identifica nell'ambito della spedizione e ha una dimensione, con codice (CDim) e descrizione (attributo Dimens.)

Con riferimento alle specifiche e ai dati forniti:

- a) mostrare le dipendenze funzionali rilevate (limitarsi a quelle che hanno a sinistra identificatori o codici):

$ID \rightarrow CM, CDes, CT$

$CM \rightarrow Mittente$

$CDes \rightarrow Des, CCD \rightarrow Categoria$ ✓

$CT \rightarrow Tipo$

$ID, NP \rightarrow CDim$

$CDim \rightarrow Dimensione$

- b) individuare la chiave (o le chiavi) della relazione;

$\{ID, NP\}^+ = CM, CDes, CT, Mittente, Des, CCD, Categoria, Tipo, CDim, Dim, ID, NP$

(ID, NP) chiave ✓

COGNOME

NOME

MATRICOLA

c) spiegare perché essa non soddisfa la BCNF;

non soddisfa BCNF perché alcune dip. funz. ^{non} contengono la chiave
 e s.x (es. $CDim \rightarrow Dim$)



d) decomporre la relazione utilizzando l'algoritmo proposto a lezione, presentando lo schema di ciascuna relazione insieme alle dipendenze funzionali associate e alle chiavi; indicare, motivandolo, quali forme normali soddisfa la decomposizione ottenuta (BCNF o 3NF);

Sped	(<u>ID</u> , C π , CDes, CT)	$ID \rightarrow C\pi, CDes, CT$
Mitt.	(<u>Cπ</u> , Mittente)	$C\pi \rightarrow Mittente$
Dest.	(<u>CDes</u> , Des, CCD, Categoria)	$CDes \rightarrow Des, CCD, Categoria$
Tipologia	(<u>CT</u> , Tipo) (<u>CCD, cot</u>)	$CT \rightarrow Tipo$
Poco	(<u>ID, NP</u> , (Dim) \rightarrow lossless join)	$ID, NP \rightarrow CDim$
Dim	(<u>CDim</u> , Dimensione)	$CDim \rightarrow Dimensione$

BCNF perché \forall dip funz. contiene la chiave della rel associata
 o s.x



e) spiegare quali proprietà (decomposizione senza perdita, preservazione delle dipendenze) sono soddisfatte dalla decomposizione ottenuta e perché.

- lossless join: non ho perdite, posso fare un join delle rel dec. e ottenere la rel originale.
- pres. dip: perché costruisco le rel sulla base delle dip funz.

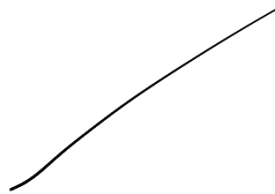


COGNOME**NOME****MATRICOLA****Esercizio 2. (REVERSE ENGINEERING)**

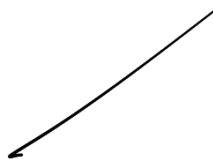
Dato il seguente schema logico relazionale

QUIZ(IdQ, *Titolo*, TempoMax_o, PuntiTotali)DOMANDA(IdQ^{QUIZ}, NumD, Testo, Punti)RISPOSTA(IdQ^{DOMANDA}, NumD^{DOMANDA}, Opzione, Testo, Percentuale, Feedback_o)STUDENTE(MatrS, Nome, Cognome)COMPILAZIONE(MatrS^{STUDENTE}, IdQ^{QUIZ}, NumTentativo, *DataOraInizio*, *DataOraFine*_o, Punteggio_o)SCELTA(MatrS^{COMPILAZIONE}, IdQ^{COMPILAZIONE,RISPOSTA}, NumTentativo^{COMPILAZIONE}, NumD^{RISPOSTA},
Opzione^{RISPOSTA})

- a) si proponga uno schema concettuale Entity Relationship la cui traduzione dia luogo a tale schema logico



- b) si modifichi lo schema per gestire il fatto che una domanda possa essere inserita in più quiz (le risposte associate sono le stesse) e possa valere punteggi diversi a seconda del quiz in cui è inserita



COGNOME

NOME

MATRICOLA

Esercizio 3. (ALGEBRA RELAZIONALE)

In riferimento al seguente schema:

QUIZ(IdQ, Titolo, TempoMaxO, PuntiTotali)

DOMANDA(IdQ^{QUIZ}, NumD, Testo, Punti)RISPOSTA(IdQ^{DOMANDA}, NumD^{DOMANDA}, Opzione, Testo, Percentuale, FeedbackO)

STUDENTE(Matrs, Nome, Cognome)

COMPILAZIONE(Matrs^{STUDENTE}, IdQ^{QUIZ}, NumTentativo, DataOraInizio, DataOraFineO, PunteggioO)SCELTA(Matrs^{COMPILAZIONE}, IdQ^{COMPILAZIONE, RISPOSTA}, NumTentativo^{COMPILAZIONE}, NumD^{RISPOSTA}, Opzione^{RISPOSTA})

Formulare le seguenti interrogazioni in algebra relazionale. Per ogni interrogazione, dopo averla formulata) effettuare i controlli richiesti e validare con V se si ritiene che il controllo sia superato, con X se si ritiene che non lo sia.

- a) Determinare gli studenti che hanno selezionato sempre e solo l'opzione "b" in tutte le domande di tutti i quiz che hanno compilato

$$\pi_{Matrs} (\sigma_{Opz = 'b'} (SCELTA))$$

$$\pi_{Matrs} (\sigma_{Opz \neq 'b'} (SCELTA))$$


- b) Determinare le compilazioni in cui il "numero tentativo" non è assegnato in maniera crescente: tale numero è precedente a quello di una compilazione dello stesso quiz avvenuta prima (=in una data e ora precedente)

$$N = \pi_{Matrs, IdQ, NumTent, DataOraInizio} (COMPILAZIONE)$$

$$\pi_{Matrs, IdQ, NumTent, DataOraInizio} (\sigma_{\substack{DataOraInizio > D \\ \wedge NumTent < NumT}} (N))$$

$$(N \bowtie \rho_{\substack{DataOraInizio \\ NumTent, IdQ \\ \leftarrow D, NumT, Q}} (N))$$

Verifica/autovalutazione	a)	b)
L'interrogazione formulata è corretta dal punto di vista dei vincoli di schema		
La richiesta e l'interrogazione formulata restituiscono una relazione con lo stesso schema		
La richiesta e l'interrogazione formulata sono entrambe monotone/non monotone		
Su una piccola istanza, la richiesta e l'interrogazione formulata restituiscono lo stesso risultato		

15420765102 11454232046 27401706422 75721021601 24307131652 43135317123 23411544745

COGNOME	NOME	MATRICOLA
---------	------	-----------

--

COGNOME	NOME	MATRICOLA
---------	------	-----------

Esercizio 4. (SQL)

In riferimento al seguente schema:

QUIZ(IdQ, Titolo, TempoMaxo, PuntiTotali)

DOMANDA(IdQ^{QUIZ}, NumD, Testo, Punti)

RISPOSTA(IdQ^{DOMANDA}, NumD^{DOMANDA}, Opzione, Testo, Percentuale, Feedbacko)

STUDENTE(MatrS, Nome, Cognome)

COMPILAZIONE(MatrS^{STUDENTE}, IdQ^{QUIZ}, NumTentativo, DataOraInizio, DataOraFineo, Punteggioo)

SCELTA(MatrS^{COMPILAZIONE}, IdQ^{COMPILAZIONE,RISPOSTA}, NumTentativo^{COMPILAZIONE}, NumD^{RISPOSTA},
Opzione^{RISPOSTA})

Formulare le seguenti interrogazioni in SQL

a) Determinare l'ultima compilazione iniziata tra quelle non ancora completate

```
SELECT MatrS, IdQ, NumTent
FROM Compilazione
WHERE DataOraFine IS NULL
AND DataOraInizio = (SELECT MAX(DataOraInizio)
FROM Compilazione)
WHERE DataOraFine IS NULL)
```

b) Determinare i quiz per cui tutti gli studenti hanno effettuato almeno un tentativo

```
SELECT IdQ
FROM Compilazione
GROUP BY IdQ
HAVING COUNT(DISTINCT MatrS)
= (SELECT COUNT(Matrs)
FROM Studenti)
```



l'architettura di riferimento coinvolge la memoria volatile e quella persistente. Della volatile consideriamo due buffer, uno riservato alle modifiche ^① delle pagine del DB, l'altro per il log ^②. Della mem. persistente consideriamo la parte dove si presenta il DB ^④ e la memoria stabile ^③ (concettualmente senza guasti). Per il protocollo WAL prima di scrivere su disco una pagina modificata ^①, viene scritto il corrispondente record di log ^② su disco ^③. Solo allora potrà scrivere la pagina modificata su disco ^④.

COGNOME

NOME

MATRICOLA

- c) Presentare uno schedule concorrente tra due transazioni che presenta l'anomalia di *letture sporche*; indicare quali livelli di isolamento permettono di evitare il verificarsi di tale anomalia, sceglierne uno e annotare lo schedule proposto con l'acquisizione, il rilascio dei lock e le attese indotte dal livello scelto, spiegando perché in questo caso l'anomalia non si potrebbe più presentare.

T1	X	T2
	o R(x)	
	o $x \leftarrow x+1$	
	1 W(x)	
R(x)	1	
	o RELEASE	
	o	
COMMIT	o	

Ha letto uno stato intermedio
→ inconsistente.

Posso evitare l'anomalia con

- SERIALIZABLE
- REPEATABLE READ
- READ COMMITTED

• READ COMMITTED

T1	X	T2
	o $x \leftarrow lock(x)$	
	o R(x)	
	o $x \leftarrow x+1$	
	1 W(x)	
S-lock(x)	1	
wait	1	
:	o RELEASE	
:	o unlock(x)	
R(x)	o	
COMMIT	o	

È facilmente osservabile che prendendo : lock, T1 non accede più a uno stato inconsistente