

COGNOME**NOME****MATRICOLA**

Basi di Dati – 20 luglio 2021

Esercizio 1. (NORMALIZZAZIONE)

a) In riferimento allo schema di relazione

PARTECIPANTE(IdP, NomeP, CognomeP, TipoScuola, Classe, Genere, Comune, Regione, Team)

formulare le dipendenze funzionali corrispondenti alle seguenti frasi in linguaggio naturale.

Ogni team è composto da partecipanti dello stesso genere e provenienti da scuole dello stesso tipo della stessa regione. Ogni comune appartiene a una regione. Ogni partecipante è identificato da un id, ha un unico nome e cognome ed è componente di un solo team.

Team \rightarrow Genere, TipoScuola, Regione

Comune \rightarrow Regione

IdP \rightarrow CognomeP, NomeP, Team

COGNOME**NOME****MATRICOLA**

- b) Data la relazione $R(A,B,C,D,E)$ e le dipendenze funzionali $C \rightarrow D$, $DE \rightarrow A$ e $AB \rightarrow C$, determinare le chiavi di R a specificare se R è in 3NF o in BCNF, motivando la risposta.

$C \rightarrow D$
 $DE \rightarrow A$
 $AB \rightarrow C$

$\{DE\}^+ = A, D, E$

$\{C\}^+ = C, D$

$\{AB\}^+ = A, B, C, D$

$\{ABE\}^+ = A, B, C, D, E$

$\{BDE\}^+ = A, B, C, D, E$

$\{BEC\}^+ = B, E, C, D, A$

di cui:

non BCNF perché due dip. non contengono di cui e > x
 3NF perché ogni dip. funz. ha a dx attributi primi.

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

Si consideri il seguente schema relazionale, relativo ad un sondaggio di tipo “doodle” utilizzato per pianificare riunioni/incontri, dove i valori possibili per Risp sono ‘sì’, ‘no’, ‘se necessario’.

SONDAGGIO (IdD, Titolo, Luogo, Descrizione, TimeZone, IdCreatore^{PARTECIPANTE})

QUANDO (IdD^{SONDAGGIO}, NumO, Data, OraI, OraF)

PARTECIPANTE (IdP, Nome, Cognome)

RISPOSTA (IdD^{QUANDO}, NumO^{QUANDO}, IdP^{PARTECIPANTE}, Risp)

NOTA (IdD^{SONDAGGIO}, IdP^{PARTECIPANTE}, TestoNota)

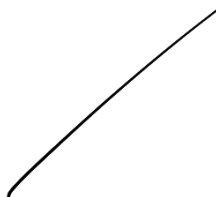
QUANDO_SCELTO (IdD^{QUANDO}, NumO^{QUANDO})

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



b) si modifichi lo schema per gestire incontri “uno a uno” (cioè per fissare appuntamenti individuali con il creatore del sondaggio) piuttosto che riunioni con più partecipanti. Questo vuol dire che

- ogni fascia oraria relativa a un certo sondaggio/riunione può essere selezionata da un solo partecipante
- ogni partecipante seleziona una sola fascia oraria relativa a un certo sondaggio/riunione
- non ci sarà più un unico orario (=quando) scelto per l’incontro



COGNOME**NOME****MATRICOLA****Esercizio 3. (ALGEBRA RELAZIONALE)**

Si consideri il seguente schema relazionale, relativo ad un sondaggio di tipo “doodle” utilizzato per pianificare riunioni/incontri, dove i valori possibili per Risp sono ‘sì’, ‘no’, ‘se necessario’.

SONDAGGIO (IdD, Titolo, Luogo, Descrizione, TimeZone, IdCreatore^{PARTECIPANTE})

QUANDO (IdD^{SONDAGGIO}, NumO, Data, OraI, OraF)

PARTECIPANTE (IdP, Nome, Cognome)

RISPOSTA (IdD^{QUANDO}, NumO^{QUANDO}, IdP^{PARTECIPANTE}, Risp)

NOTA (IdD^{SONDAGGIO}, IdP^{PARTECIPANTE}, TestoNota)

QUANDO_SCELTO (IdD^{QUANDO}, NumO^{QUANDO})

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 i nomi dei creatori dei sondaggi che hanno sia opzioni (=quando) in orari pomeridiani (cioè con OraI > 13) che mattutini (cioè OraF < 13)

$\pi_{IdP, Nome} (\sigma_{OraI > 13} (QUANDO \bowtie RISPOSTA \bowtie PART))$

\cap

$\pi_{IdP, Nome} (\sigma_{OraI < 13} (QUANDO \bowtie RISPOSTA \bowtie PART))$



- b) Determinare i partecipanti che hanno risposto almeno un ‘sì’ a tutti i sondaggi

$\pi_{IdD, IdP} (\sigma_{Risp = 'sì'} (RISPOSTA))$

$\div \pi_{IdP} (PARTECIPANTE)$



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		

COGNOME**NOME****MATRICOLA****Esercizio 4. (SQL)**

Si consideri il seguente schema relazionale, relativo ad un sondaggio di tipo “doodle” utilizzato per pianificare riunioni/incontri, dove i valori possibili per Risp sono ‘sì’, ‘no’, ‘se necessario’.

SONDAGGIO (IdD, Titolo, Luogo, Descrizione, TimeZone, IdCreatore^{PARTECIPANTE})

QUANDO (IdD^{SONDAGGIO}, NumO, Data, OraI, OraF)

PARTECIPANTE (IdP, Nome, Cognome)

RISPOSTA (IdD^{QUANDO}, NumO^{QUANDO}, IdP^{PARTECIPANTE}, Risp)

NOTA (IdD^{SONDAGGIO}, IdP^{PARTECIPANTE}, TestoNota)

QUANDO_SCELTO (IdD^{QUANDO}, NumO^{QUANDO})

Formulare le seguenti interrogazioni in SQL.

- a) Determinare il sondaggio contenente la fascia oraria (=quando) con l’ora di fine più tarda

```

SELECT ASINAT IDD
FROM QUANDO
WHERE OraF = (SELECT MAX (OraF)
              FROM QUANDO)
  
```



- a) Determinare per ogni fascia oraria (=quando) proposta dei sondaggi per cui non è ancora stato scelto l’orario, il numero di risposte ‘sì’ ottenute

```

SELECT COUNT (*)
FROM QUANDO
WHERE Risp = 'sì' AND OraI IS NULL
GROUP BY IDD, NumO
  
```



15420765102 11454232046 27401706422 75721021601 24307131652 43135317123 23411544745

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

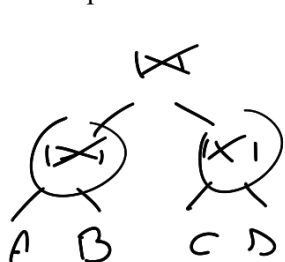
COGNOME

NOME

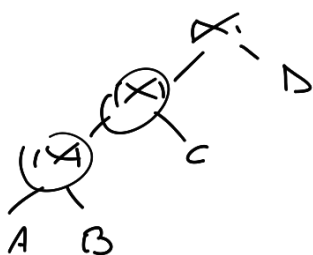
MATRICOLA

PARTE III. DOMANDE, SOLO PER 12 CFU

- a) Discutere e confrontare la modalità di esecuzione di un operatore in modalità materializzata o in pipeline, presentando almeno un esempio per ciascuna modalità.



nested loop
 join materializzato, devo salvare i risultati intermedi di almeno un join per poter effettuare i confronti tuple per tuple con tutte le tuple della relazione inner (prodotto da una dei due join)



nested loop
 join in pipeline, sfruttando il piccolo left deep le relazioni inner e sempre materializzate, ma i risultati intermedi (cardinalità) dei join non vanno materializzati.

- b) Descrivere, utilizzando opportuno esempi, almeno due anomalie che si possono presentare durante l'esecuzione di uno schedule concorrente.

- lost update: viene perso una modifica del DB, il risultato è visibile nel momento ad esempio di un acquisto: l'ultimo prezzo viene acquistato da due persone diverse
- dirty read: lettura sporca = lettura di uno stato intermedio e cui segue ad esempio un rollback.
 Ad es. legge la presenza di un biglietto per una data ma al momento dell'acquisto tale data non esiste più.

COGNOME

NOME

MATRICOLA

- c) Presentare un piano di esecuzione logico e un corrispondente piano di esecuzione fisico, per una interrogazione di vostra scelta.

SELECT A.k
FROM A
WHERE A.k = 2

Lqp

 π_k $\sum_{k=2}$

A

 $\pi_k(\sum_{k=2}(A))$

pgp: se ho indice su A.k posso sfruttare una scansione su indice. Se questo è di hash ancora meglio, in quanto vi escede in tempo costante (presupponendo no overflow). Ciò per implementare la selezione.
Se non avessi indice, eseguo una scansione sequenziale.

Per la proiezione posso effettuare una scelta iterativa; se voglio rimuovere i duplicati posso ordinare prima il mio file di dati (caso da lavorare: duplicati occorrono).