

**COGNOME****NOME****MATRICOLA****Basi di Dati – 14 giugno 2021****Esercizio 1. (NORMALIZZAZIONE)**

Considerare la relazione seguente:

Cod	Titolo	CC	Città	CN	Nazione	CM	Materia	CA	Area
1	Femore	C1	Firenze	N1	Italia	M1	Ortopedia	A1	Medicina
2	Ginocchio	C2	Roma	N1	Italia	M1	Ortopedia	A1	Medicina
3	Cuore2018	C1	Firenze	N1	Italia	M2	Cardiologia	A1	Medicina
4	DB2018	C1	Berlino	N2	Germania	M3	Basi di dati	A2	Informatica

La relazione mostra (in forma non normalizzata) i dati di interesse per un insieme di convegni scientifici, secondo le seguenti specifiche:

- ogni convegno ha un codice e un titolo e si tiene una città;
- ogni città ha un nome, si trova in una nazione e ha un codice che la identifica nell'ambito della nazione;
- ogni nazione ha un codice e un nome;
- ogni convegno è relativo ad una materia specifica (ad esempio l'ortopedia) che a sua volta appartiene a un'area scientifica (ad esempio la medicina);
- ogni materia ha un nome e un codice che la identifica univocamente;
- ogni area scientifica ha un codice e un nome.

Con riferimento alle specifiche e ai dati forniti:

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

$Cod \rightarrow Titolo, CC, CN, CM$

$CC, CN \rightarrow Città$

$CN \rightarrow Nazione$

$CM \rightarrow Materia, CA$

$CA \rightarrow Area$

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

$\{Cod\}^+ = Cod, Titolo, CC, CN, CM, Città, Nazione, Materia, CA, Area$

- c) spiegare perché essa non soddisfa la BCNF;

Non tutte le dip. funz. contengono la chiave a sinistra.

COGNOME

NOME

MATRICOLA

- d) mostrare una decomposizione della relazione originaria che soddisfi la BCNF (mostrare le tabelle, con i dati, indicando le dipendenze associate a ciascuna tabella e la chiave);

Convegno ( Cod , Titolo, CC, CN, CM )

$Cod \rightarrow \text{Titolo, CC, CN, CM}$

Luogo ( CC, CN , Città )

$CC, CN \rightarrow \text{Città}$

Luogo N ( CN , Nozione )

$CN \rightarrow \text{Nozione}$

Tema ( CM , Motore, CA )

$CM \rightarrow \text{Motore, CA}$

Interesse ( CA , Area )

$CA \rightarrow \text{Area}$

Chiave contenuta in almeno una rel. dec.  $\rightarrow$  lossless join

A dip. funz. associate alle rel. decomposte, le parte sx contiene una chiave



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

lossless join  $\rightarrow$  facendo il join delle rel. dec. ottengo la rel. originale;  
ottenibile perché almeno una rel. dec. contiene la chiave originale

preserva le dip.  $\rightarrow$  soddisfa perché associato ad ogni rel. la sua dip. funz.



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

## Esercizio 2. (REVERSE ENGINEERING)

Dato il seguente schema logico relazionale

SEDE(IdSede, NomeSede, Indirizzo, CAP, Comune)

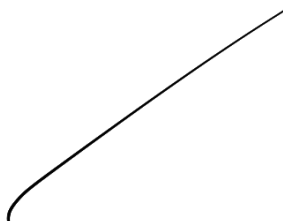
DIPENDENTE(Email, Nome, Cognome, Qualifica, Telefono, IdSede<sup>SEDE</sup>, Dipartimento<sup>DIPARTIMENTO</sup>)

DIPARTIMENTO(Sigla, Nome, Direttore<sup>DIPENDENTE</sup>) Nome e Direttore sono entrambe chiavi alternative (due chiavi, ciascuna costituita da un attributo)

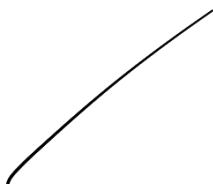
AULA(IdAula, IdSede<sup>SEDE</sup>, NomeAula<sub>o</sub>, TipoAula, Capienza, Accessibile, Dipartimento<sup>DIPARTIMENTO</sup>)

PRENOTAZIONE(IdAula<sup>AULA</sup>, IdSede<sup>AULA</sup>, Data, OraI, OraF, Descrizione, TipoAttività, PartecipantiAttesi, PrenotataDa<sup>DIPENDENTE</sup>, ConfermataDa<sup>DIPENDENTE</sup>)

- 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 singola prenotazione (che sarà ora identificata da un id), fatta da un dipendente e (eventualmente) confermata da un dipendente, relativa a un'attività (di un certo tipo, in una certa data, ora inizio e ora fine, e partecipanti attesi), possa coinvolgere più aule [nella stessa sede o in sedi diverse]



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

### Esercizio 3. (ALGEBRA RELAZIONALE)

In riferimento al seguente schema:

SEDE(IdSede, NomeSede, Indirizzo, CAP, Comune)

DIPENDENTE(Email, Nome, Cognome, Qualifica, Telefono, IdSede<sup>SEDE</sup>)

AULA(IdAula, IdSede<sup>SEDE</sup>, NomeAula, TipoAula, Capienza, Accessibile)

PRENOTAZIONE(IdAula<sup>AULA</sup>, IdSede<sup>AULA</sup>, Data, OraI, OraF, Descrizione, TipoAttività, PartecipantiAttesi, PrenotataDa<sup>DIPENDENTE</sup>, ConfermataDa<sup>DIPENDENTE</sup>)

Formulare le seguenti interrogazioni in algebra relazionale

- a) Determinare le aule di capienza superiore a 100 della sede (il cui nome è) Villa Bonino in cui non sono mai state effettuate prenotazioni per attività di tipo lezione

$$\pi_{IdAula, IdSede} \left( \sigma_{CAP > 100 \wedge namesede = 'Villa Bonino'} (AULA \bowtie SEDE) \right)$$

$$- \pi_{IdAula, IdSede} \left( \sigma_{TipoAtt = 'lezione'} (PRENOTAZIONE) \right)$$

- b) Determinare le email dei dipendenti che hanno effettuato prenotazioni di aule per attività di **tutti** i tipi nella sede (il cui nome è) Villa Bonino

$$\pi_{PRENOTADA, TipoAtt} \left( \sigma_{NomeSede = 'Villa Bonino'} (SEDE \bowtie PRENOTAZIONE) \right)$$

$$\frac{\sigma}{\pi} \pi_{TipoAtt} (PRENOTAZIONE)$$

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

#### Esercizio 4. (SQL)

In riferimento al seguente schema:

SEDE(IdSede, NomeSede, Indirizzo, CAP, Comune)

DIPENDENTE(Email, Nome, Cognome, Qualifica, Telefono, IdSede<sup>SEDE</sup>)

AULA(IdAula, IdSede<sup>SEDE</sup>, NomeAula, TipoAula, Capienza, Accessibile)

PRENOTAZIONE(IdAula<sup>AULA</sup>, IdSede<sup>AULA</sup>, Data, OraI, OraF, Descrizione, TipoAttività, PartecipantiAttesi, PrenotataDa<sup>DIPENDENTE</sup>, ConfermataDa<sup>DIPENDENTE</sup>)

#### Formulare le seguenti interrogazioni in SQL

- a) Determinare per ogni sede il numero di prenotazioni effettuate per aule di tale sede, quante aule diverse sono state prenotate e il numero medio di partecipanti attesi

```
SELECT IdSede, COUNT(*), COUNT(DISTINCT IDAULA), AVG(partAtt)
FROM Prenotazione
GROUP BY IdSede
```



- b) Determinare le aule la cui capienza è inferiore alla capienza media delle aule dello stesso tipo

```
SELECT IdAula, IdSede
FROM Aula AS A
WHERE Capienza < (SELECT AVG(Capienza)
FROM Aula
WHERE TipoAula = A.TipoAula)
```



15420765101 11454232046 27401706422 75721021601 24307131652 43135317123 23411544745

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

COGNOME

NOME

MATRICOLA

## PARTE III. DOMANDE, SOLO PER 12 CFU

- a) Descrivere il livello di isolamento REPEATABLE READ, specificando quando devono essere acquisiti e rilasciati i lock condivisi ed esclusivi secondo lo standard, quali anomalie permette di evitare e perché. Usare un esempio se lo si ritiene opportuno.

Questo livello di isolamento permette solo anomalie phantom row, impedendo lost update, dirty read e unrepeatable read.  
 Segue il protocollo di locking Strong 2PL. Con questo livello la transazione legge solo modifiche effettuate da altre transazioni che hanno fatto commit e tali valori non vengono modificati da altri finché questa non ha terminato.

- b) Descrivere le politiche force e no-force utilizzate nell'implementazione del protocollo Write-Ahead-Logging.

Quando una transazione finisce nello stato committed, si possono seguire due opzioni circa la scrittura su disco delle pagine nel buffer prima o dopo la scrittura della tupla "commit" nel log.

**force:** si forza immediatamente la scrittura su disco delle pagine nel buffer e solo successivamente si scrive commit nel log.  
 Di contro, se ha molte modifiche della stessa pagina ha molte scritture. In caso di questo no redo.

**no force:** si ritarda la scrittura su disco a quando è possibile e successivamente. Ciò migliora gli accessi a disco: scrivono solo quando il buffer deve essere liberato.  
 In caso di questo sono force REDO.

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

c) Descrivere il processo di elaborazione delle interrogazioni, illustrandone i vari passi.

La query in SQL viene compilata e tradotta in un parsetree (AST) o più. Di questi vengono scritte le espressioni algebriche canoniche per poi passare all'ottimizzazione logica. Seguendo delle euristiche viene scelto il LQP (piano logico ottimizzato) migliore che viene passato all'ottimizzatore fisico che ne costruisce un piano fisico ottimizzato (PQP). Infine viene elaborato ed eseguito il piano, portando a termine l'elaborazione dell'interrogazione.