

## Prima Prova in Itinere 6 dicembre 2023 – Prova A

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente database relativo alla gestione dei teatri, con relative opere, stagioni ed abbonamenti:

Opera (**id**, titolo, durata)

Stagione (**teatro**, **anno**, **opera**, costo\_abbonamento)

Teatro (**id**, nome, citta, numero\_posti)

Abbonamento (**persona**, **teatro**, **anno**)

Persona (**id**, nome, cognome, indirizzo, citta, cap, provincia)

1. Identificare le chiavi primarie ed esterne dello schema (per le chiavi esterne indicare anche a quale relazione e attributo fa riferimento ogni chiave) [1 punto];

2. Rispondere alle seguenti query in algebra relazionale:

- a. Per ogni anno, trovare il teatro (restituendo nome e luogo) in cui si è tenuta l'opera con maggiore durata [3 punti];

$$R_2 = R_1 = \pi_{teatro, anno, opera, durata}(stagione \bowtie_{id=opera} opera)$$

$$R_3 = \pi_{R_1.teatro, R_1.anno, R_1.opera, R_1.durata}(R_1 \bowtie_{R_1.anno=R_2.anno \wedge R_1.durata < R_2.durata} R_2)$$

$$\pi_{nome, citta, anno, opera, durata}((R_1 - R_3) \bowtie_{teatro=id} Teatro)$$

- b. Trovare le persone che nella stagione 2023 si sono abbonate solo in teatri della propria città [4 punti];

$$R_1 = \pi_{id, nome, cognome, citta, teatro} \left( Persona \bowtie_{id=persona} \sigma_{anno=2023}(Abbonamento) \right)$$

$$R_2 = \pi_{id, nome, cognome, citta, teatro}(R_1 \bowtie_{teatro=id \wedge R_1.citta <> Teatro.citta} Teatro)$$

$$R_1 - R_2$$

3. Rispondere alle seguenti query in SQL:

**Prima Prova in Itinere 6 dicembre 2023 – Prova B**

Matricola	Nome e Cognome	Valutazione

- a. Trovare i teatri con un numero totale di abbonati della stessa città maggiore rispetto alla media degli abbonati nei teatri delle altre città [3 punti];

```
CREATE VIEW ABBONATITEATRO AS (  
  SELECT COUNT(*) abbonati, Teatro  
  FROM abbonamento  
  GROUP BY teatro)  
  
SELECT COUNT(*) abbonatiLocali, teatro  
FROM Persona, Teatro, abbonamento  
WHERE Persona.id = abbonamento.persona AND  
Teatro.id=abbonamento.teatro AND Persona.citta = Teatro.citta  
GROUP BY teatro  
HAVIGN abbonatiLocali > (SELECT AVG(abbonati) FROM  
ABBONATITEATRO)
```

- b. Creare un vincolo che non permetta di inserire un nuovo abbonamento per una stagione di un teatro se non ci sono più posti disponibili [4 punti].

```
CREATE ASSERTION A CHECK (  
  NOT EXISTS SELECT *  
    FROM TEATRO t  
    WHERE numero_posti < ANY (SELECT COUNT(*)  
      FROM ABBONAMENTO  
      WHERE Teatro=t.id  
      GROUP BY anno))
```

## Prima Prova in Itinere 5 dicembre 2022 – Prova C

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente database relativo alla gestione del registro insegnamento dei corsi universitari:

Materia (**ID**, titolo, descrizione, anno\_accademico, CFU, ore, completato)

CorsoDiLaurea (**id**, titolo, classe)

MaterieCorsoDiLaurea (**corso\_di\_studi**, **materia**)

Insegnamento (**id**, **materia**, professore)

Registro (**materia**, data, numero\_ore, contenuto)

- Identificare le chiavi primarie ed esterne dello schema (per le chiavi esterne indicare anche a quale relazione e attributo fa riferimento ogni chiave) [1 punto];
- Rispondere alle seguenti query in algebra relazionale:
  - Per ogni docente identificare le materie che insegna con il numero di CFU più alto, mostrare il nome del docente, il nome della materia e il titolo del corso di studi [3 punti];

$$R_2 = R_1 = \pi_{professore, id, titolo, CFU} (insegnamento \bowtie materia)$$

$$R_3 = \pi_{R_1.*} (R_1 \bowtie_{R_1.professore=R_2.professore \wedge R_1.CFU < R_2.CFU} R_2)$$

$$(R_1 - R_3) \bowtie \text{completare con la join opportuna}$$

- Identificare i docenti che nell'A.A. 22/23 hanno insegnano solamente nel corso di studi in "Medicina" [4 punti];

$$R_1 = \pi_{professore} (Insegnamento \bowtie \sigma_{anno_{accademico}=22/23} (materia) \bowtie materieCordoDi Laurea \bowtie \sigma_{titolo <> "Medicina"} (corsoDiLaurea))$$

$$R_2 = \pi_{professore} (Insegnamento \bowtie \sigma_{anno_{accademico}=22/23} (materia) \bowtie materieCordoDi Laurea)$$

$$R_2 - R_1$$

- Rispondere alle seguenti query in SQL:

- Visualizzare nome e cognome dei docenti che nell'A.A. 22/23 hanno tenuto un numero ore effettivo maggiore di quello medio tenuto da loro stessi negli anni precedenti [3 punti];

```
CREATE VIEW oreeffettive AS (
SELECT sum(numero_ore) oreins, professore, anno_accademico
FROM registro, insegnamento, materia
WHERE registro.materia = insegnamento.materia AND
insegnamento.id=materia,id AND anno_accademico<>"22/23"
```

**Prima Prova in Itinere 6 dicembre 2023 – Prova B**

Matricola	Nome e Cognome	Valutazione

```
GROUP BY professore,anno_accademico)
```

```
CREATE VIEW ore2223 AS (  
SELECT sum(numero_ore) oreinsegnamento, professore  
FROM registro, insegnamento, materia  
WHERE registro.materia = insegnamento.materia AND  
insegnamento.id=materia,id AND anno_accademico="22/23  
GROUP BY professore)
```

```
SELECT professore  
FROM ore223 o  
Where ore.oreinsegnamento > (SELECT AVG(oreins)  
FROM oreeffettive  
Where professore=o.professore)
```

- b. Implementare un trigger che a raggiungimento del numero di ore previsto per una materia, aggiorna il campo "completato" di materia a TRUE [4 punti].

```
CREATE TRIGGER CorsoCompleto  
AFTER insert on Registro  
When (SELECT sum(numero_ore) FROM registro WHERE  
materia=new.materia =  
SELECT ore FROM materia WHERE materia.id =new.materia)  
UPDATE materia SET completato = TRUE where id=new.materia
```