

Prima Prova in Itinere 5 dicembre 2022 – Prova A

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente database relativo agli eventi sportivi per atleti di varie categorie:

Atleta (id, nome, cognome, datanascita, genere)

Categoria (id, nome, descrizione)

CategoriaPartecipante (partecipante, categoria, datainizio, datafine)

Evento (id, data, luogo)

Piazzamento (evento, persona, piazzamento)

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. Trovare gli atleti che hanno partecipato ad eventi in tutti i luoghi [3 punti];
 - b. Trovare gli atleti (restituendo nome, cognome, e genere) che si sono sempre piazzati per ultimi in eventi della città di Catania [4 punti];
3. Rispondere alle seguenti query in SQL:
 - a. Trovare gli eventi con un numero di partecipanti nati dopo il 2010 maggiore rispetto alla media dei partecipanti degli eventi nello stesso luogo [3 punti];
 - b. Creare un vincolo che non permetta di inserire due piazzamenti con la stessa posizione per persone diverse dello stesso evento [4 punti].

```
CREATE ASSERTION EvitaPiazzamentiDuplicati
CHECK (NOT EXIST (
    SELECT * FROM Piazzamento p1, Piazzamento p2
    WHERE p1.evento = p2.evento AND p1.piazzamento = p2.piazzamento
    AND p1.persona <> p2.persona
))
```

CdL in Informatica Triennale – A.A. 2022-2023

Basi di Dati

Proff. S. Alaimo - A. Pulvirenti

Prima Prova in Itinere 5 dicembre 2022 – Prova B

Matricola	Nome e Cognome	Valutazione

```
CREATE TRIGGER EvitaPiazzamentiDuplicati
BEFORE INSERT ON Piazzamento
FOR EACH ROW
DECLARE conteggio NUMBER
BEGIN
    SELECT COUNT(*) INTO conteggio
    FROM Piazzamento
    WHERE evento=NEW.evento
    AND piazzamento=NEW.piazzamento;

    IF (conteggio > 0) THEN
        SIGNAL SQLSTATE '1000001' 'Piazzamento già utilizzato';
    END IF
END
```

Prima Prova in Itinere 5 dicembre 2022 – Prova C

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente database relativo alla gestione delle certificazioni linguistiche:

Studente (**matricola**, nome, cognome, datanascita)

Livelli (**id**, nome, descrizione, numero_livello)

Certificazioni (**id**, titolo, descrizione, durata_certificazione, livello)

CertificazioneStudente (**certificazione**, **studente**, **data**, valutazione, esito)

Si consideri, inoltre, che: la durata della certificazione è espressa in **mesi**; l'esito della certificazione può avere i valori **POSITIVO** o **NEGATIVO**; la valutazione della certificazione è compresa **tra 0 e 10**.

- 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 certificazione trovare gli studenti che hanno avuto la più alta valutazione [3 punti];

$$R_1 = R_2 = \pi_{studente, certificazione, valutazione}(CertificazioneStudente)$$

$$R_3 = \pi_{R_1.studente, R_1.certificazione} (R_1 \bowtie_{R_1.valutazione < R_2.valutazione \wedge R_1.certificazione = R_2.certificazione} R_2)$$

$$\pi_{studente, certificazione}(R_1) - R_3$$

- Trovare le certificazioni a cui hanno partecipato tutti gli studenti nati prima del 1990 [4 punti];

$$R_1 = \pi_{matricola}(\sigma_{datanascita < "1990-01-01"}(Studente))$$

$$\delta_{studente \rightarrow matricola}(\pi_{certificazione, studente}(CertificazioneStudente)) \div R_1$$

- Rispondere alle seguenti query in SQL:
 - Trovare gli studenti che hanno ottenuto con esito positivo tutte le certificazioni della lingua turca (cercare turco nel titolo delle certificazioni) [3 punti];

```
SELECT *
FROM Studente s
WHERE NOT EXIST (
    SELECT * FROM Certificazioni c
    WHERE c.titolo LIKE '%turco%'
    AND NOT EXIST (
```

Prima Prova in Itinere 5 dicembre 2022 – Prova B

Matricola	Nome e Cognome	Valutazione

```
SELECT * FROM CertificazioneStudente cs
WHERE cs.certificazione = c.id
AND cs.studente = s.matricola
AND cs.esito = 'POSITIVO')
```

)

- b. Creare un trigger che non permetta l’inserimento di un esito positivo in una certificazione di un livello se non si è conseguita con successo una certificazione del livello inferiore [4 punti].

Prima Prova in Itinere 5 dicembre 2022 – Prova C

Matricola	Nome e Cognome	Valutazione

```
CREATE TRIGGER ControllaLivelloCertificazione
BEFORE INSERT ON CertificazioneStudente
FOR EACH ROW
DECLARE nome VARCHAR
DECLARE livello, conteggio NUMBER

BEGIN
    SELECT c.titolo, l.numero_livello INTO nome, livello
    FROM Certificazioni c, Livelli l
    WHERE l.id = c.livello
    AND c.id = NEW.certificazione;

    SELECT COUNT(*) INTO conteggio
    FROM CertificazioneStudente cs, Certificazioni c, Livelli l
    WHERE cs.studente = NEW.studente AND cs.esito='POSITIVO'
    AND cs.certificazione = c.id AND c.livello = l.id
    AND c.titolo = nome AND l.numero_livello = livello - 1;
    IF (conteggio = 0) THEN

        SIGNAL SQLSTATE '000001' 'Non ci sono certificazioni del livello
precedente'

    END IF
END
```

Prima Prova in Itinere 5 dicembre 2022 – Prova B

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente database relativo alla gestione delle coltivazioni dei terreni agricoli in un comune:

Terreno (**id**, foglio, numero, sub, dimensione)

Coltivazioni (**id**, nome, descrizione)

ColtivazioniTerreno (**id**, **terreno**, **coltivazione**, dettagli)

Raccolto (**coltivazione**, **anno**, esito, quotazione)

Si consideri, inoltre, che: la dimensione del terreno è espressa in **ettari**; l'esito della coltivazione può avere i valori **POSITIVO** o **NEGATIVO**; la quotazione è espressa con un numero intero in **centesimi di euro**.

- 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:
 - Trovare i terreni più piccoli che hanno avuto le più alte quotazioni di raccolto nel 2022 [3 punti];

$$R_1 = R_2 = \text{Terreno}$$

$$R_3 = \pi_{R_1.id} (R_1 \bowtie_{R_1.dimensione > R_2.dimensione} R_2)$$

$$R_4 = \pi_{R_1.id} (R_1) - R_3$$

$$R_5 = R_4 \bowtie_{R_4.id = terreno} \text{ColtivazioneTerreno} \bowtie_{Raccolto.coltivazione = ColtivazioneTerreno.id} \text{Raccolto}$$

$$R_6 = R_7 = \pi_{terreno, quotazione} (\sigma_{anno=2022} (R_5))$$

$$R_8 = \pi_{R_6.terreno} (R_6 \bowtie_{R_6.quotazione < R_7.quotazione} R_7)$$

$$\pi_{terreno} (R_6) - R_8$$

Prima Prova in Itinere 5 dicembre 2022 – Prova C

Matricola	Nome e Cognome	Valutazione

- b. Trovare i terreni in cui sono praticate tutte le coltivazioni, indicando tutti i dati del terreno [4 punti];

$$R_1 = \delta_{id \rightarrow coltivazione}(\pi_{id}(Coltivazione))$$

$$R_2 = \pi_{terreno, coltivazione}(ColtivazioneTerreno) \div R_1$$

$$R_2 \bowtie_{R_2.terreno=Terreno.id} Terreno$$

3. Rispondere alle seguenti query in SQL:

- a. Trovare i terreni che non hanno avuto raccolti con esito positivo per coltivazioni di Colza del 2021 [3 punti];

```
SELECT *
FROM Terreno t
WHERE NOT EXIST (

    SELECT *
    FROM ColtivazioneTerreno ct, Coltivazione c, Raccolto r
    WHERE ct.coltivazione = c.id
    AND r.coltivazione = ct.id
    AND ct.terreno = t.id AND c.nome = "Colza"
    AND r.anno = 2021 AND r.esito = "POSITIVO"

)
```

- b. Creare un vincolo che non permetta di inserire le stesse coltivazioni in un terreno per più di 4 (quattro) anni [4 punti].

Prima Prova in Itinere 5 dicembre 2022 – Prova B

Matricola	Nome e Cognome	Valutazione

ColtivazioniTerreno (**id**, terreno, coltivazione, dettagli)

Raccolto (**coltivazione**, anno, esito, quotazione)

```
CREATE TRIGGER ControllaInserimentoColtivazione
AFTER INSERT ON ColtivazioniTerreno
FOR EACH ROW
DECLARE conteggio NUMBER
BEGIN

    SELECT COUNT(*) INTO conteggio
    FROM Raccolto r, ColtivazioniTerreno c
    WHERE r.coltivazione = c.id
    AND c.coltivazione = NEW.coltivazione;

    IF (conteggio > 4) THEN
        DELETE FROM ColtivazioniTerreno WHERE id=NEW.id;
    END IF

END
```