

Basi di Dati e Sistemi informativi aziendali

Prova scritta di esame - 2019/20

2 - SQL (*Testo, soluzioni, esempio, errori frequenti*)

Produrre le interrogazioni SQL relative ai tre esercizi riportati qui sotto a partire dal database descritto.

Database

Database - Specifiche

Il database da utilizzare per la prova contiene le informazioni relative a **sciatori di sci di fondo** (chiamato anche *sci nordico*) e alle **gare disputate**.

Sciatore (CodSc, NomeSc, Categoria, Nazione)..... gli sciatori di fondo

CodSc codice dello sciatore

NomeSc..... nome dello sciatore

Categoria categoria dello sciatore (*allievi, juniores, seniores, master, ...*)

Nazione..... nazione di appartenenza dello sciatore

Gara (NomeGara, Localita, Specialita, Distanza)..... le gare disputate

NomeGara nome della gara

Localita..... località di svolgimento della gara

Specialita..... specialita della gara (*tecnica classica, tecnica libera, ...*)

Distanza lunghezza della gara in km

Partecipa (CodSc, NomeGara, Posizione)..... la partecipazione degli sciatori alle gare

CodSc codice dello sciatore (*chiave esterna alla tabella Sciatore*)

NomeGara nome della gara (*chiave esterna alla tabella Gara*)

Posizione..... posizione nella classifica generale della gara (1, 2, , 3 ...)

Database di esempio - Dati

Skiatore				
N	<u>CodSc</u>	<u>NomeSc</u>	<u>Categoria</u>	<u>Nazione</u>
	SMALLINT	VARCHAR(64)	VARCHAR(64)	VARCHAR(64)
1	1	Bolshunov	Master	Russia
2	2	Klaebo	Master	Norvegia
3	3	Niskanen	Master	Finlandia
4	4	Graz	Juniore	Italia
5	5	Del Fabbro	Juniore	Italia
6	6	Moch	Juniore	Germania
7	7	Cologna	Seniore	Svizzera
8	8	Di Centa	Seniore	Italia
9	9	Milz	Seniore	Germania

Gara				
N	<u>NomeGara</u>	<u>Localita</u>	<u>Specialita</u>	<u>Distanza</u>
	VARCHAR(64)	VARCHAR(64)	VARCHAR(64)	SMALLINT
1	Marcialonga	Fassa	Tecnica classica	70
2	Kangaroo	Fassa	Tecnica libera	85
3	Juniorelauf	Davos	Tecnica classica	25
4	Dolomitenlauf	Davos	Tecnica libera	50
5	Vasaloppet	Falun	Tecnica classica	90

Partecipa			
N	<u>CodSc</u>	<u>NomeGara</u>	<u>Posizione</u>
	SMALLINT	VARCHAR(64)	SMALLINT
1	1	Marcialonga	10
2	2	Marcialonga	2
3	3	Marcialonga	3
4	4	Marcialonga	4
5	5	Marcialonga	5
6	6	Marcialonga	6
7	7	Marcialonga	7
8	8	Marcialonga	8
9	9	Marcialonga	9
10	1	Kangaroo	9
11	2	Kangaroo	8
12	3	Kangaroo	10
13	4	Kangaroo	1
14	5	Kangaroo	5
15	6	Kangaroo	4
16	7	Kangaroo	9
17	8	Kangaroo	7
18	9	Kangaroo	8
19	4	Juniorelauf	4
20	5	Juniorelauf	5
21	1	Dolomitenlauf	7
22	2	Dolomitenlauf	6
23	3	Dolomitenlauf	5
24	6	Dolomitenlauf	4
25	7	Dolomitenlauf	3
26	8	Dolomitenlauf	2
27	9	Dolomitenlauf	1
28	4	Vasaloppet	1
29	5	Vasaloppet	2
30	6	Vasaloppet	3

Database di esempio - SQL per creazione e popolamento

Importante: tutti gli esempi, le istruzioni SQL e le considerazioni che seguono sono riferiti al DBMS MySQL

```
SET storage_engine = InnoDB;

CREATE DATABASE IF NOT EXISTS Appello_Feb2;
USE Appello_Feb2;

DROP TABLE IF EXISTS Sciatore;
CREATE TABLE Sciatore (
    CodSc SMALLINT,
    NomeSc VARCHAR(64),
    Categoria VARCHAR(64),
    Nazione VARCHAR(64),
    PRIMARY KEY(CodSc)
);

DELETE FROM Sciatore;
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(1, 'Bolshunov', 'Master', 'Russia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(2, 'Klaebo', 'Master', 'Norvegia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(3, 'Niskanen', 'Master', 'Finlandia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(4, 'Graz', 'Juniores', 'Italia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(5, 'Del Fabbro', 'Juniores', 'Italia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(6, 'Moch', 'Juniores', 'Germania');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(7, 'Cologna', 'Seniores', 'Svizzera');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(8, 'Di Centa', 'Seniores', 'Italia');
INSERT INTO Sciatore(CodSc, NomeSc, Categoria, Nazione) VALUES(9, 'Milz', 'Seniores', 'Germania');

DROP TABLE IF EXISTS Gara;
CREATE TABLE Gara (
    NomeGara VARCHAR(64),
    Localita VARCHAR(64),
    Specialita VARCHAR(64),
    Distanza SMALLINT,
    PRIMARY KEY(NomeGara)
);

DELETE FROM Gara;
INSERT INTO Gara(NomeGara, Localita, Specialita, Distanza) VALUES('Marcialonga', 'Fassa', 'Tecnica classica', 70);
INSERT INTO Gara(NomeGara, Localita, Specialita, Distanza) VALUES('Kangaroo', 'Fassa', 'Tecnica libera', 85);
INSERT INTO Gara(NomeGara, Localita, Specialita, Distanza) VALUES('Junioreslauf', 'Davos', 'Tecnica classica', 25);
INSERT INTO Gara(NomeGara, Localita, Specialita, Distanza) VALUES('Dolomitenlauf', 'Davos', 'Tecnica libera', 50);
INSERT INTO Gara(NomeGara, Localita, Specialita, Distanza) VALUES('Vasaloppet', 'Falun', 'Tecnica classica', 90);

DROP TABLE IF EXISTS Partecipa;
CREATE TABLE Partecipa (
    CodSc SMALLINT,
    NomeGara VARCHAR(64),
    Posizione SMALLINT,
    PRIMARY KEY(CodSc, NomeGara)
);

DELETE FROM Partecipa;
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(1, 'Marcialonga', 10);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(2, 'Marcialonga', 2);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(3, 'Marcialonga', 3);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(4, 'Marcialonga', 4);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(5, 'Marcialonga', 5);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(6, 'Marcialonga', 6);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(7, 'Marcialonga', 7);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(8, 'Marcialonga', 8);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(9, 'Marcialonga', 9);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(1, 'Kangaroo', 9);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(2, 'Kangaroo', 8);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(3, 'Kangaroo', 10);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(4, 'Kangaroo', 1);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(5, 'Kangaroo', 5);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(6, 'Kangaroo', 4);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(7, 'Kangaroo', 9);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(8, 'Kangaroo', 7);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(9, 'Kangaroo', 8);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(4, 'Junioreslauf', 4);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(5, 'Junioreslauf', 5);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(1, 'Dolomitenlauf', 7);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(2, 'Dolomitenlauf', 6);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(3, 'Dolomitenlauf', 5);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(6, 'Dolomitenlauf', 4);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(7, 'Dolomitenlauf', 3);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(8, 'Dolomitenlauf', 2);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(9, 'Dolomitenlauf', 1);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(4, 'Vasaloppet', 1);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(5, 'Vasaloppet', 2);
INSERT INTO Partecipa(CodSc, NomeGara, Posizione) VALUES(6, 'Vasaloppet', 3);
```

Esercizio 1

Produrre l'elenco (**NomeSc**, **AvgPosizione**, **TotDistanza**) dove ogni riga contiene i seguenti attributi:

- **NomeSc**: il nome di uno sciatore
- **AvgPosizione**: la media delle posizioni ottenute dallo sciatore calcolata sulle gare con distanza maggiore o uguale a 30 km
- **TotDistanza**: la somma delle distanze percorse dallo sciatore, sempre calcolata sulle gare con distanza maggiore o uguale a 30 km

L'elenco deve essere

- *limitato* agli sciatori che hanno percorso in totale (**TotDistanza**) almeno 200 km
- *ordinato* in senso *crescente* per **AvgPosizione** e *decrescente* per **TotDistanza**.

Soluzione

```
SELECT NomeSc, AVG(Posizione) AS AvgPosizione, SUM(Distanza) AS TotDistanza
FROM Sciatore S, Gara G, Partecipa P
WHERE S.CodSc = P.CodSc AND
      P.NomeGara = G.NomeGara AND
      Distanza >= 30
GROUP BY NomeSc
HAVING TotDistanza >= 200
ORDER BY AvgPosizione, TotDistanza DESC;
```

Risultato nel database di esempio

NomeSc	AvgPosizione	TotDistanza
Graz	2,000	245
Del Fabbro	4,000	245
Moch	4,250	295
Klaebo	5,333	205
Bolshunov	5,667	205
Di Centa	5,667	205
Milz	6,000	205
Niskanen	6,000	205
Cologna	6,333	205

Errori frequenti

Omissione della clausola GROUP BY NomeSc

- Senza questa clausola il risultato della query è del tutto casuale in quanto tutte le righe selezionate dalla clausola **WHERE**, non essendo raggruppate per nome sciatore, confluiscono in un'unica riga; in questa riga, in particolare, NomeSc è un valore random (dipendente dall'ordine di popolamento della tabella) e i termini **AVG(Posizione)**, **SUM(Distanza)** sono il risultato dell'accorpamento di tutti i concorrenti di tutte le gare in un unico valore del tutto privo di significato.

Nota: in DBMS non MySQL l'omissione della clausola **GROUP BY** comporta generalmente un errore di sintassi

Inclusione della condizione TotDistanza >= 200 **nella clausola** WHERE

```
WHERE S.CodSc = P.CodSc AND
      P.NomeGara = G.NomeGara AND
      Distanza >= 30 AND
      TotDistanza >= 200
```

Poiché TotDistanza è il risultato di una funzione di aggregazione, deve essere inserito in una clausola di tipo **HAVING** posta dopo **GROUP BY**.

Esercizio 2

Produrre l'elenco (**NomeGara, Localita**) delle gare con specialità "Tecnica classica" a cui hanno partecipato solo sciatori italiani di categoria "Juniores"

Soluzione

```
SELECT NomeGara, Localita
FROM Gara
WHERE Specialita = 'Tecnica classica' AND
NomeGara NOT IN (
    SELECT NomeGara
    FROM Sciatore S, Partecipa P
    WHERE S.CodSc = P.CodSc AND
        (Nazione <> 'Italia' OR Categoria <> 'Juniores')
);
```

Risultato nel database di esempio

NomeGara	Localita
Junioreslauf	Davos

Errori frequenti

Selezione delle gare a cui hanno partecipato anche sciatori italiani "Juniores"

- Ognuna delle due soluzioni che seguono (del tutto equivalenti) restituiscono le gare dove hanno partecipato anche sciatori italiani juniores, ma non escludono che, nelle stesse gare, abbiano partecipato altri sciatori (stranieri, italiani non juniores, ecc.)

```
SELECT NomeGara, Localita
FROM Gara
WHERE Specialita = 'Tecnica classica' AND NomeGara IN (
    SELECT NomeGara
    FROM Sciatore S, Partecipa P
    WHERE S.CodSc = P.CodSc AND
        (Nazione = 'Italia' AND Categoria = 'Juniores')
);
```

```
SELECT DISTINCT G.NomeGara, Localita
FROM Gara G, Sciatore S, Partecipa P
WHERE G.NomeGara = P.NomeGara AND
    P.CodSc = S.CodSc AND
    Specialita = 'Tecnica classica' AND
    Nazione = 'Italia' AND
    Categoria = 'Juniores';
```

Selezione delle gare a cui possono aver partecipato sciatori italiani non "Juniores" oppure sciatori non italiani "Juniores"

- La soluzione che segue è già migliore rispetto alla precedente, ma, per errore, esclude solo le gare con sciatori contemporaneamente non italiani e non juniores, potendo invece comprendere sciatori italiani non juniores oppure sciatori juniores non italiani (che dovrebbero invece essere esclusi).

```
SELECT NomeGara, Localita
FROM Gara
WHERE Specialita = 'Tecnica classica' AND
NomeGara NOT IN (
    SELECT NomeGara
    FROM Sciatore S, Partecipa P
    WHERE S.CodSc = P.CodSc AND
        (Nazione <> 'Italia' AND → OR
        Categoria <> 'Juniores')
);
```

Esercizio 3

Produrre l'elenco (**Categoria, NomeSc, Posizione**) dove ogni riga contiene i seguenti attributi:

- **Categoria**: una categoria di sciatore
- **NomeSc**: il nome dello sciatore appartenente a quella categoria che, nella gara di nome "Marcialonga", è arrivato *primo nella sua categoria* (suggerimento: "primo di categoria" *non equivale* a "primo nella classifica generale", ma comporta che la posizione di quello sciatore sia *la minima* tra gli sciatori appartenenti alla sua stessa categoria)
- **Posizione**: la posizione dello sciatore nella classifica generale (sempre della *Marcialonga*).

Soluzione

```
SELECT  Categoria, NomeSc, Posizione
FROM    Sciatore S1, Partecipa P
WHERE   S1.CodSc = P.CodSc AND
        NomeGara = 'Marcialonga' AND
        Posizione = (
            SELECT  MIN(Posizione)
            FROM    Sciatore S2, Partecipa P
            WHERE   S2.CodSc = P.CodSc AND
                    NomeGara = 'Marcialonga' AND
                    S1.Categoria = S2.Categoria
        );
```

Risultato nel database di esempio

Categoria	NomeSc	Posizione
Master	Klaebo	2
Juniors	Graz	4
Seniors	Cologna	7

Errori frequenti

Mancanza della sottoproposizione per individuare i primi concorrenti di ogni categoria

- In alcuni casi è stata proposta, come risultato, una query analoga alla seguente:

```
SELECT  Categoria, NomeSc, Posizione
FROM    Sciatore S, Partecipa P
WHERE   S.CodSc = P.CodSc AND
        NomeGara = 'Marcialonga'
GROUP BY Categoria
HAVING  Posizione = MIN(Posizione);
```

Il risultato di questa query è del tutto casuale in quanto il raggruppamento

GROUP BY Categoria restituisce un concorrente random per ciascuna categoria (e non necessariamente il primo classificato della categoria); pertanto, se questo concorrente non è il primo della sua categoria, si ha **Posizione <> MIN(Posizione)** e la query non restituisce alcun concorrente di quella categoria.

Nota: in DBMS non MySQL la query sopra riportata comporta generalmente un errore di sintassi.