

Indirizzo: ITIA - INFORMATICA E TELECOMUNICAZIONI  
ARTICOLAZIONE INFORMATICA

©Giselda De Vita 2023

La traccia di esame, pubblicata dal Ministero, si trova al seguente link: [Traccia 2023](#)

## Soluzione seconda parte

**Quesito I - In relazione al tema proposto nella prima parte, si sviluppi, in un linguaggio a scelta, una porzione di codice significativa delle pagine web necessarie a presentare la classifica generale degli studenti di una certa classe virtuale, in base alle monete raccolte in tutti i videogiochi di quella classe.**

Il quesito chiede di inserire all'interno di un pezzo di codice la classifica degli studenti ma, a differenza della query sviluppata al punto 4 b), si dovranno considerare tutti i videogiochi.

Per la form supponiamo solo di dover selezionare la classe da una combobox. Selezionata la classe apparirà la classifica.

Modifichiamo la query 4 punto b) per far apparire anche il videogame giocato.

```
SELECT videogiochi.titolo , nome, cognome, monete
FROM Studenti, videogiochi, premi, studente_classe
WHERE premi.id_studente = Studenti.id
AND premi.cod_vodeogioco = videogiochi.codice
AND studente_classe.id_studente = Studenti.id
AND studente_classe.cod_classe = 'AAABBB'
ORDER BY videogiochi.titolo, monete DESC;
```

Form html per la combo. Al submit verrà chiamato il codice PHP.

## Classifica per classe

Classe

3B - MATEMATICA

Crea

```
<html>
  <head>
    <title>Esame 2023</title>
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/
3.3.6/css/bootstrap.min.css">
  </head>
  <body>

    <form class="form-horizontal" action="selclassifica2023.php" method="post" >
      <fieldset>

        <!-- Form Name -->
        <legend>Classifica per classe</legend>

        <!-- Select Basic -->
        <div class="form-group">
          <label class="col-md-4 control-label" for="classe">Classe</label>
          <div class="col-md-4">
            <select id="classe" name="classe" class="form-control">
              <option value="AAABBB">3B - MATEMATICA</option>
              <option value="BBBAAA">4C - INFORMATICA</option>
              <option value="XVYQEJ">4C - STORIA</option>
            </select>
          </div>
        </div>

        <!-- Button -->
        <div class="form-group">
          <label class="col-md-4 control-label" for=""></label>
          <div class="col-md-4">
            <button id="" name="" class="btn btn-primary">Crea</button>
          </div>
        </div>
      </fieldset>
    </form>

  </body>
</html>
```

## Codice PHP selclassifica2023.php

```
<html>
<head>
  <title>Esempio traccia 2023 – quesito uno seconda parte</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/
dist/css/bootstrap.min.css">
  <!-- esempio bootstrap per le table al seguente indirizzo:
  https://getbootstrap.com/docs/4.0/content/tables/
  prendiamo come esempio la tabella a righe alternate colorate "Striped rows"
-->
</head>
<body>
  <?
  $classe=$_POST['classe'];

  echo "<h1>Classifica STUDENTI classe " . $classe . "</h1>";

  $conn = new mysqli("localhost","root","", "2023");
  if ($conn->connect_error) {
    die("Connessione fallita con errore: " . $conn->connect_error);
  }

  $query = "SELECT videogiochi.titolo as titolo , nome, cognome, monete ".
  " FROM Studenti, videogiochi, premi, studente_classe ".
  " WHERE premi.id_studente = Studenti.id ".
  " AND premi.cod_vodeogioco = videogiochi.codice ".
  " AND studente_classe.id_studente = Studenti.id ".
  " AND studente_classe.cod_classe = '". $classe. "' ".
  " ORDER BY videogiochi.titolo, monete DESC";

  $result = $conn->query($query);

  $contatorePosizione = 1; //posizione in classifica
  $videogame = ''; //videogame relativo alla classifica

  if ($result->num_rows > 0) {
    echo "<table class='table table-striped'>";

    echo "<thead><tr><th>Posizione</th><th>Titolo</th><th>Studente</th>";
    echo "<th>Monete</th></tr></thead> <tbody>";

    while($row = $result->fetch_assoc()) {
      if($videogame == $row['titolo']) //stesso videogame incremento la
classifica
        {$contatorePosizione++;}
      else{ //cambiato videogame reietto classifica
        $videogame = $row['titolo'];
        $contatorePosizione = 1;
      }

      echo "<tr> <th>".$contatorePosizione;
      echo "</th><td>".$row['titolo'];
      echo "</td><td>".$row['nome'] . " " . $row['cognome'] ;
      echo "</td><td>".$row['monete']. "</td></tr>";
    }
  }
}
```

```

    }
} else {
    echo "<h3>Nessun studente ha accumulato monete</h3>";
}
echo "</tbody> </table>";
$conn->close();
?>
</body>
</html>

```

Output del seguente codice:

## Classifica STUDENTI classe AAABBB

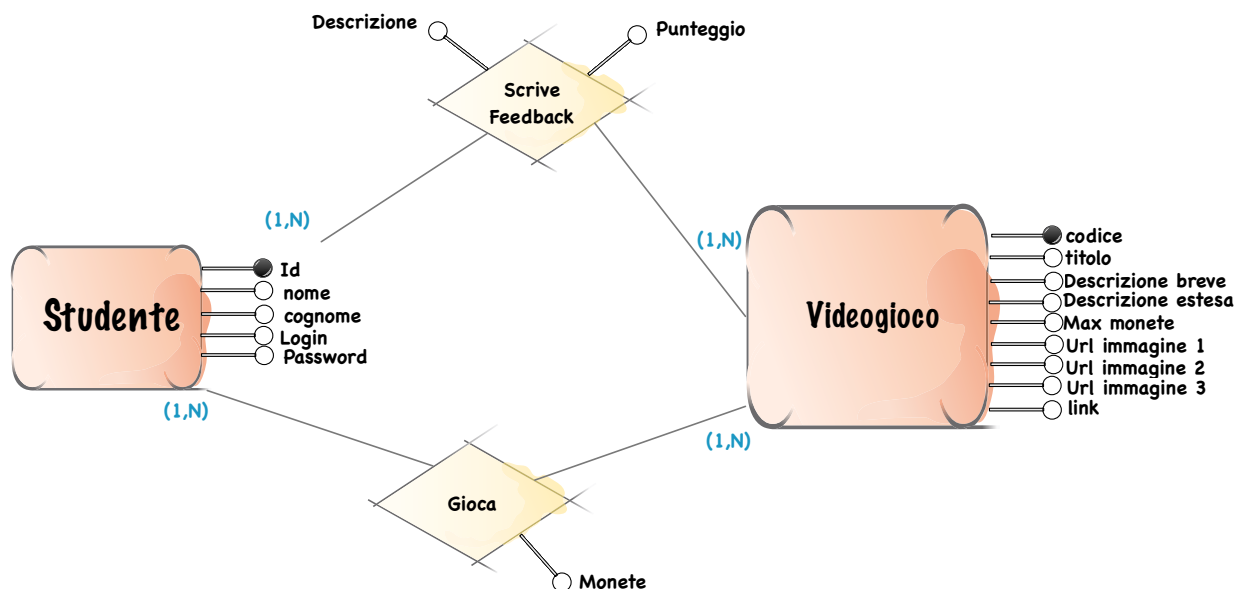
Posizione	Titolo	Studente	Monete
1	PITAGORIAMO	Studente uno COGNOME	10
2	PITAGORIAMO	Studente uno COGNOME	5
3	PITAGORIAMO	Studente DUE COGNOME2	2
1	TABELLIAMO	Studente DUE COGNOME2	8
2	TABELLIAMO	Studente 3 Cognome 3	4
3	TABELLIAMO	Studente uno COGNOME	3

### Osservazioni

Esempio classico di SELECT tramite codice PHP. Unica difficoltà era il riformulare la query ed in seguito trovare un piccolo stratagemma per calcolare per la posizione. Tramite un contatore che si resetta ogni volta che cambia videogame, si riesce ad indicare anche la posizione in classifica.

**Quesito 2 - In relazione al tema proposto nella prima parte, si descriva in che modo è possibile integrare la base di dati sopra sviluppata, per gestire anche i feedback da parte degli studenti sui videogiochi. Ogni feedback è costituito da un punteggio che può andare da 1 a 5 e una descrizione di massimo 160 caratteri. Si descriva anche la struttura delle pagine web dedicate a tale funzionalità, scrivendo in un linguaggio a scelta una porzione di codice significativa di tali pagine.**

Prendendo solo le entità studente e video gioco, inseriamo la relazione per effettuare un feedback.



Uno studente scrive recensioni su più Videogiochi, un VideoGioco viene recensito da più studenti, per cui la relazione è molti a molti.

Si traduce nel passaggio al logico con una tabella **feedback** con i seguenti attributi:

**Feedback** (punteggio, descrizione, id\_studente, codice\_videogioco);

```

CREATE TABLE Feedback (
    id_studente int(4) NOT NULL,
    codice_videogioco int(5) NOT NULL,
    descrizione varchar(160) NOT NULL,
    punteggio int(1) NOT NULL,
    PRIMARY KEY (id_studente, codice_videogioco)
);
    
```

Il problema grafico di rappresentare un rating con delle stelline per il gradimento è un classico del mondo web.

Tra tantissimi esempi, su w3school c'è il pezzetto di codice HTML e CSS per visualizzare le stelline al seguente indirizzo: [https://www.w3schools.com/howto/howto\\_css\\_star\\_rating.asp](https://www.w3schools.com/howto/howto_css_star_rating.asp)

Di seguito un esempio semplicissimo per effettuare una query sul database e visualizzare le stelline come emoji attraverso il codice Unicode per HTML.

Se il punteggio è 4 con un for fino al valore del punteggio si stampano le stelline gialle, dal punteggio fino a 5 si stampano le stelline bianche bordo nero.

```
<?php
$studente=$_POST['id_studente'];
$videogame =$_POST['cod_videogame'];

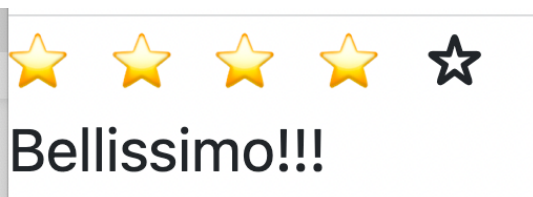
$conn = new mysqli("localhost","root","", "2023");
if ($conn->connect_error) {
    die("Connessione fallita con errore: " . $conn->connect_error);
}

$query = "SELECT descrizione, punteggio FROM Feedback WHERE id_studente = ".
$studente." AND codice_videogioco = ".$videogame;

$result = $conn->query($query);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc(); //una sola riga per un determinato studente
    //un videogioco
    //un videogioco
    for($i=0; $i < $row['punteggio'];$i++)
        echo '<b> ★</b> &nbsp;';
    for($i = $row['punteggio']; $i < 5; $i++)
        echo ' ★&nbsp;';
    echo '<p> '. $row['descrizione'] . '</p>';
}
?>
```

Output



### Quesito 3 - Si descriva, anche attraverso esempi, il concetto di “raggruppamento” nelle interrogazioni SQL, indicando in tale contesto come operano le funzioni di aggregazione e la clausola HAVING.

La clausola **GROUP BY** raggruppa le righe che la clausola WHERE seleziona in base al valore della colonna/e di raggruppamento indicate.

La clausola **HAVING** è un altro filtro che si applica ai campi indicati dalla clausola **GROUP BY** e si applica solo ai gruppi che soddisfano la condizione di ricerca, scartando il resto.

Nel caso del nostro database gli esempi si sprecano ma si deve ricordare che l'applicazione più semplice del **GROUP BY**, e relativa clausola **HAVING**, è accanto alla funzione di raggruppamento come COUNT, MIN, MAX, SUM e AVG.

### ESEMPI

Essendo le funzioni di raggruppamento particolarmente indicate per effettuare conteggi ed operazioni su campi di tipo numerico, pensiamo di voler trovare la MEDIA delle monete guadagnate dagli studenti di una determinata classe.

```
SELECT studente_classe.cod_classe, videogiochi.titolo , AVG(monete)
FROM Studenti, videogiochi, premi, studente_classe
WHERE premi.id_studente = Studenti.id
AND premi.cod_vodeogioco = videogiochi.codice
AND studente_classe.id_studente = Studenti.id
GROUP BY videogiochi.titolo, studente_classe.cod_classe
HAVING studente_classe.cod_classe = 'AAABBB';
```

Un altro esempio più semplice è ricavare il numero di studenti registrati all'interno di ogni classe ma si vogliono visualizzare solo le classi quarte.

```
SELECT COUNT( Studenti.id), studente_classe.cod_classe
FROM Studenti, studente_classe, classi
WHERE studente_classe.id_studente = Studenti.id
AND classi.codice = studente_classe.cod_classe
GROUP BY classi.nome
HAVING classi.nome like '4%';
```

**Quesito 4 - Data la seguente tabella “Progetti”, il candidato verifichi se soddisfa le proprietà di normalizzazione e proponga uno schema relazionale equivalente che rispetti la terza Forma Normale, motivando le scelte effettuate. Si implementi in linguaggio SQL lo schema relazionale ottenuto**

ID	Titolo	Budget	Tipo	DataInizio	DataFine	Tutor	TelTutor
1	Pensiero computazionale	40.000	PON	20/02/2023	Null	Rossi Mario	345678910
2	Robotica educativa	13.000	PCTO	10/11/2022	30/03/2023	Bianchi Carlo	333444555
3	Tinkering	25.000	PCTO	14/10/2022	20/02/2023	Bianchi Carlo	333444555
4	Realtà virtuale	30.000	PCTO	16/02/2023	30/05/2023	Rossi Mario	345678910

In questa tabella sono presenti diverse anomalie, ridondanza di dati e anomalie di cancellazione e modifica. Ad esempio, modificando un numero di telefono del tutor si dovrà aggiornare lo stesso dato su diverse righe. Problema classico che si risolve con la NORMALIZZAZIONE.

Procediamo per step ricordando le varie forme normali.

### Prima forma NORMALE

*Esiste una chiave primaria e ogni attributo è definito su un dominio di valori atomici, ovvero ogni attributo è semplice quindi non composto.*

Analizzando campo per campo ci imbattiamo nel TUTOR che contiene sia nome che cognome.

Rendiamo gli attributi atomici, nome tutor e cognome tutor.

ID	Titolo	Budget	Tipo	DataInizio	DataFine	Tutor nome	Tutor cognome	TelTutor
1	Pensiero computazionale	40.000	PON	20/02/2023	Null	Mario	Rossi	345678910
2	Robotica educativa	13.000	PCTO	10/11/2022	30/03/2023	Carlo	Bianchi	333444555
3	Tinkering	25.000	PCTO	14/10/2022	20/02/2023	Carlo	Bianchi	333444555
4	Realtà virtuale	30.000	PCTO	16/02/2023	30/05/2023	Mario	Rossi	345678910



Nella tabella progetti esiste una chiave primaria, il campo ID per i progetti ma non sappiamo se il tutor dallo stesso nome e cognome sia la stessa persona oppure un caso di omonimia. Aggiungiamo quindi una chiave primaria codice per il tutor (potrebbe essere anche il codice fiscale) e presupponiamo che siano la stessa persona.

ID	Titolo	Budget	Tipo	DataInizio	DataFine	Cod Tutor	Tutor nome	Tutor cognome	TelTutor
1	Pensiero computazionale	40.000	PON	20/02/2023	Null	T01	Mario	Rossi	345678910
2	Robotica educativa	13.000	PCTO	10/11/2022	30/03/2023	T02	Carlo	Bianchi	333444555
3	Tinkering	25.000	PCTO	14/10/2022	20/02/2023	T02	Carlo	Bianchi	333444555
4	Realtà virtuale	30.000	PCTO	16/02/2023	30/05/2023	T01	Mario	Rossi	345678910

## Seconda forma normale

*Un tabella è in seconda forma normale se è in prima forma normale e se tutti i suoi attributi sono dipendenti dall'intera chiave primaria.*

Essendo le chiavi ID progetto e Cod Tutor, appare evidente che alcuni attributi sono dipendenti solo da una o dall'altra chiave per cui dovremo dividere in due tabelle aggiungendo l'opportuna chiave esterna per non perdere il collegamento tra progetti e tutor.

ID	Titolo	Budget	Tipo	DataInizio	DataFine	Cod Tutor
1	Pensiero computazionale	40.000	PON	20/02/2023	Null	T01
2	Robotica educativa	13.000	PCTO	10/11/2022	30/03/2023	T02
3	Tinkering	25.000	PCTO	14/10/2022	20/02/2023	T02
4	Realtà virtuale	30.000	PCTO	16/02/2023	30/05/2023	T01

Cod Tutor	Tutor nome	Tutor cognome	TelTutor
T01	Mario	Rossi	345678910
T02	Carlo	Bianchi	333444555

## Terza forma normale

Una tabella e' in terza forma normale se e' in seconda forma normale e tutti gli attributi, che non fanno parte della chiave primaria, dipendono esclusivamente dalla chiave stessa.

Per quanto riguarda la tabella tutor tutti gli attributi dipendono dalla chiave primaria del tutor per cui possiamo affermare che è in terza forma normale. Per quanto riguarda i progetti, per essere precisi, notiamo la colonna Tipo che non è propriamente dipendente dalla chiave primaria indicando una tipologia. Se la descrizione PCTO o PON dovesse cambiare, oppure se ci servisse la dicitura per intero, preferiamo creare una terza tabella **Tipo** che conterrà un codice e sarà quello a legarsi alla tabella progetto slegandolo così dalla descrizione.

ID	Titolo	Budget	Tipo	DataInizio	DataFine	Cod Tutor
1	Pensiero computazionale	40.000	1	20/02/2023	Null	T01
2	Robotica educativa	13.000	2	10/11/2022	30/03/2023	T02
3	Tinkering	25.000	2	14/10/2022	20/02/2023	T02
4	Realtà virtuale	30.000	2	16/02/2023	30/05/2023	T01

Cod Tutor	Tutor nome
1	PON
2	PCTO

Cod Tutor	Tutor nome	Tutor cognome	TelTutor
T01	Mario	Rossi	345678910
T02	Carlo	Bianchi	333444555

```
CREATE TABLE Tutor (  
  Codice varchar(3) NOT NULL,  
  nome varchar(128) NOT NULL,  
  Cognome varchar(128) NOT NULL,  
  telefono varchar(24) NOT NULL,  
  PRIMARY KEY (Codice));
```

```
CREATE TABLE Progetti (  
  id int(4) NOT NULL,  
  titolo varchar(128) NOT NULL,  
  budget int(6) NOT NULL,  
  tipo int(2) NOT NULL,  
  Data_inizio date NOT NULL,  
  Data_fine date,  
  cot_tutor varchar(3) NOT NULL,  
  PRIMARY KEY (id));
```

```
CREATE TABLE tipo (  
  codice int(2) NOT NULL,  
  descrizione varchar(64) NOT NULL,  
  PRIMARY KEY (codice));
```