

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](#)

La traccia d'esame di quest'anno, ritenuta dagli studenti lineare e fattibile, chiede di realizzare per una scuola una piattaforma web per la fruizione di educational games (videogiochi in ambito educativo), per migliorare l'apprendimento nelle varie materie.

La traccia è semplice perchè ha eliminato la complessità di architetture distribuite caratteristiche delle ultime tracce di informatica o informatica/sistemi.

Quesito 1 - un'analisi della realtà di riferimento, giungendo alla definizione di uno schema concettuale della base di dati che, a suo motivato giudizio, sia idoneo a gestire la realtà presentata;

Trattandosi di una piattaforma interna alla scuola, possiamo anche fermarci ad una sicurezza di primo livello (*login e password*) perchè non verranno trattati dati sensibili dei docenti e degli alunni, ma solo il progresso sulla piattaforma didattica.

Come suggerisco sempre agli studenti, leggendo la traccia individuiamo, evidenziandoli, i soggetti candidati a diventare le entità nel nostro modello concettuale.

Disciplina: INFORMATICA

Il candidato svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

PRIMA PARTE

Una scuola vuole progettare una piattaforma web per la fruizione di *educational games* (ovvero videogiochi in ambito educativo), per migliorare l'apprendimento nelle varie materie.

Ciascun **docente**, una volta completata la registrazione alla piattaforma, può creare una o più **classi** virtuali (identificate da un nome e una materia di pertinenza: es. 3B, matematica) e aprire l'iscrizione alle singole classi ai propri studenti tramite la condivisione del codice iscrizione (link o QR-code).

Nella piattaforma è presente il catalogo dei **videogiochi** didattici, classificati in base ad un elenco di **argomenti** prestabiliti (es: triangoli, legge di Ohm, verismo ...): ciascun docente può selezionare uno o più videogiochi per includerli in una classe virtuale. Per ogni videogioco è presente un titolo, una descrizione breve di massimo 160 caratteri, una descrizione estesa, il numero di "monete virtuali" che si possono raccogliere all'interno del gioco e fino a tre immagini sul gioco.

Uno **studente** si iscriverà sulla piattaforma alle classi cui è stato invitato (es: 3B matematica, 3B italiano ...) tramite il relativo codice iscrizione, e all'interno di ciascuna classe troverà i link ai videogiochi didattici proposti dal docente. Svolgendo ciascun videogioco, lo studente potrà raccogliere sequenzialmente delle **monete** tramite quiz o attività da completare. Una moneta è un riconoscimento che viene assegnato nel videogioco al raggiungimento di determinati traguardi educativi graduali.

Attraverso il numero di monete, raccolte man mano da uno studente in ciascun videogioco di quella classe, si può determinare una classifica per ciascun gioco e anche una classifica generale comprensiva di tutti i giochi della classe; il docente può quindi seguire l'andamento degli studenti e supportarli individualmente nel completamento della raccolta delle monete.

Ricordiamo di dover sottolineare tutti i sostantivi che potrebbero essere dei candidati per diventare delle entità all'interno del nostro database.

ATTENZIONE alle NON ENTITA'

Scuola non va evidenziata perché è il committente. Il sistema informativo sarà all'interno della scuola non entra l'interno del database.

La **piattaforma**, nominata più volte, è tutto il sistema informativo che stiamo realizzando per cui non va all'interno del database.

La **classifica** non è un'entità perché è una raccolta di dati presenti all'interno del database. Come si evidenzia dalla richiesta numero 4 punto b) in cui viene proprio richiesta la classifica come risultato di una query.

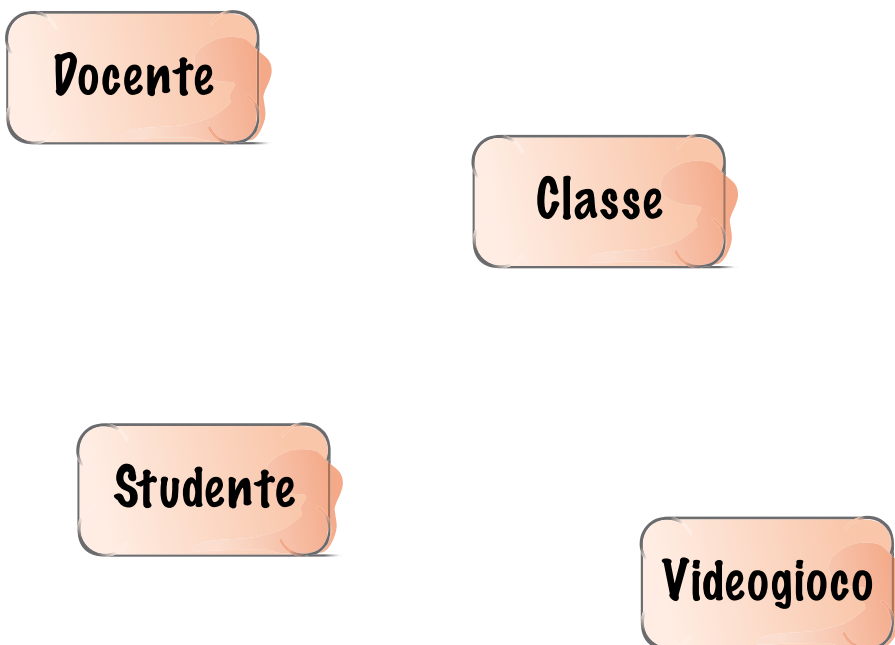
Il **catalogo** dei videogiochi sarebbe molto utile come vista ma essendo un insieme di videogiochi vale lo stesso ragionamento fatto per classifica.

Grande attenzione a quei sostantivi che sono gli attributi di una entità. Li riconosciamo facilmente perché da soli non hanno un significato e vengono sempre accompagnati dalla domanda: "Di chi? Di cosa?"

Facendo un esempio pratico, uno studente è uno studente e non ha necessità di nessun'altra definizione o di un'ulteriore specifica. D'altro canto il sostantivo **descrizione** capiamo subito che si accompagna alla domanda: "Descrizione di cosa?"; ed infatti dovremo aggiungere che, all'interno della traccia, la descrizione breve o estesa si riferisce al video gioco educativo.

ENTITA'

Prime entità forti e sicuramente presenti all'interno del database:



All'interno della traccia abbiamo selezionato i candidati a diventare le nostre entità nel modello concettuale:

- **Docente**
- **Studente**
- **Classe**
- **Videogioco**

Entità deboli

Da una seconda lettura della traccia possiamo individuare alcune entità più deboli che rientreranno nel disegno del nostro database.

Argomento relativo ai video giochi. Di fatto l'entità argomento categoria è un classico in genere si preferisce inserirlo come entità, piuttosto che come attributo complesso, per la facilità nella modifica e nell'inserimento di un nuovo argomento. Nel caso in cui si scegliesse di trattarlo come un attributo complesso, nel momento in cui si inserisce un nuovo argomento oppure si volesse modificare la descrizione di un argomento, si dovrà intervenire proprio sulla struttura della tabella, modificando la definizione dell'attributo complesso. Per cui è molto più corretto inserire una entità che permetta una facile modifica o inserimento o cancellazione di nuovi argomenti.

Le **Monete** nominate più volte ma esistono solo nel momento in cui uno studente li guadagna vincendo o superando un determinato video gioco. Capiamo subito che si tratta di un'entità molto debole perché è semplicemente un contatore che non ha senso di esistere se non all'interno di una relazione tra studente e videogioco. Per questo motivo non rientrerà all'interno del modello concettuale ma di sicuro nel nostro database ci sarà la possibilità di memorizzare le monete che uno studente accumula giocando ai vari videogiochi.

Attributi

Rileggiamo la traccia selezionando stavolta quelli che appaiono come attributi delle nostre entità:

Disciplina: INFORMATICA

Il candidato svolga la prima parte della prova e due tra i quesiti proposti nella seconda parte.

PRIMA PARTE

Una scuola vuole progettare una piattaforma web per la fruizione di *educational games* (ovvero videogiochi in ambito educativo), per migliorare l'apprendimento nelle varie materie.

Ciascun **docente**, una volta completata la registrazione alla piattaforma, può creare una o più **classi** virtuali (identificate da un **nome** e una **materia di pertinenza**: es. 3B, matematica) e aprire l'iscrizione alle singole classi ai propri studenti tramite la condivisione del **codice iscrizione** (link o QR-code).

Nella piattaforma è presente il catalogo dei **videogiochi** didattici, classificati in base ad un elenco di **argomenti** prestabiliti (es: triangoli, legge di Ohm, verismo ...): ciascun docente può selezionare uno o più videogiochi per includerli in una classe virtuale. Per ogni videogioco è presente un **titolo**, una **descrizione breve** di massimo 160 caratteri, una **descrizione estesa**, il **numero** di "monete virtuali" che si possono raccogliere all'interno del gioco e fino a **tre immagini** sul gioco.

Uno **studente** si iscriverà sulla piattaforma alle classi cui è stato invitato (es: 3B matematica, 3B italiano ...) tramite il relativo **codice iscrizione**, e all'interno di ciascuna classe troverà i **link** ai videogiochi didattici proposti dal docente. Svolgendo ciascun videogioco, lo studente potrà raccogliere sequenzialmente delle **monete** tramite **quiz o attività** da completare. Una moneta è un riconoscimento che viene assegnato nel videogioco al raggiungimento di determinati traguardi educativi graduali.

Attraverso il **numero** di monete, raccolte man mano da uno studente in ciascun videogioco di quella classe, si può determinare una classifica per ciascun gioco e anche una classifica generale comprensiva di tutti i giochi della classe; il docente può quindi seguire l'andamento degli studenti e supportarli individualmente nel completamento della raccolta delle monete.

Dalla traccia non vengono indicati gli attributi di docenti e studenti che manterremo al minimo indispensabile per il funzionamento della piattaforma essendo il sistema interno alla scuola.

Docente - ID, e-mail, nome, cognome, login, password

Studente - ID, nome, cognome, login, password

Classe - Codice iscrizione, nome, materia di pertinenza, link iscrizione

Argomento - codice, descrizione

Videogioco - codice, titolo, descrizione breve, descrizione estesa, numero massimo di monete, link, tre immagini.

Osservazioni:

Per quanto riguarda la **classe**, immaginiamo che il *codice di iscrizione* sia una sequenza di caratteri alfanumerici univoci da poter inserire magari in una pagina che richiede il codice di iscrizione, candidata ideale per essere la chiave primaria. Si aggiunge il **link di iscrizione**, che può essere inviato agli studenti, inteso come

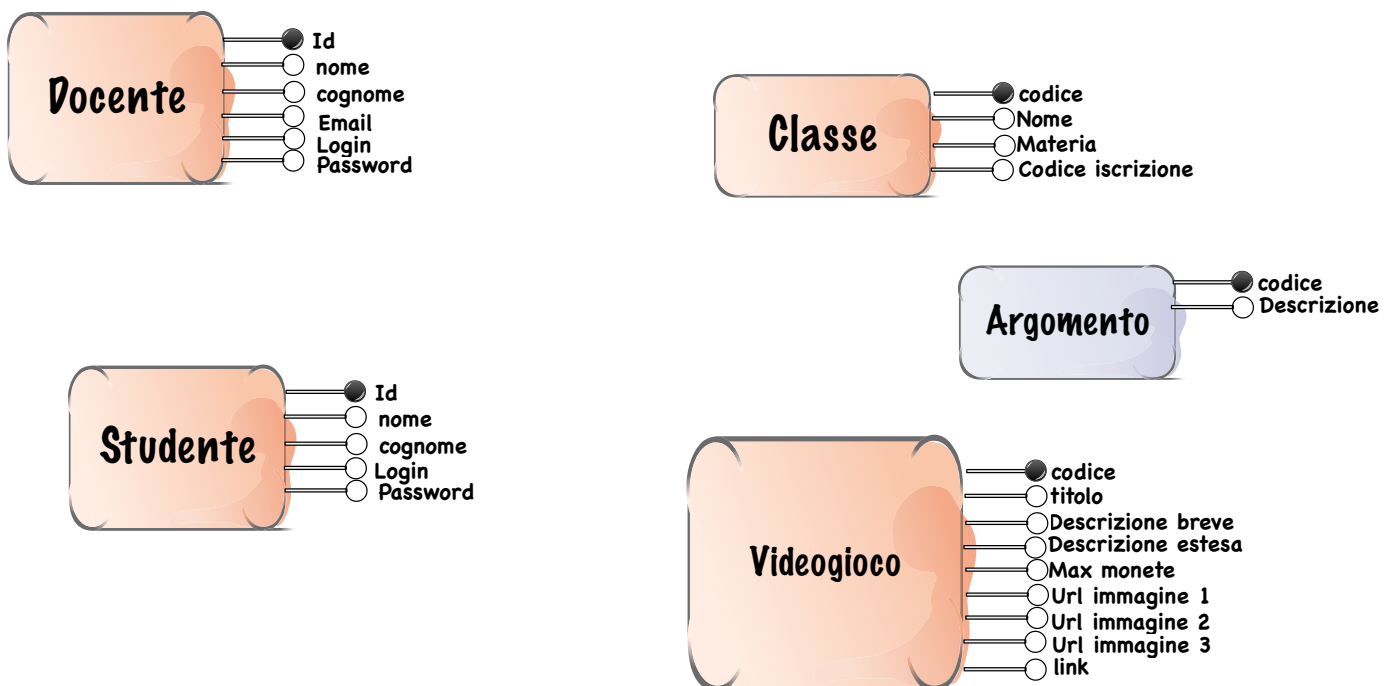
URL completo. Sulla creazione random di una sequenza di caratteri è stato implementato il punto 6 in PHP.

Per quanto riguarda il QR code non entra all'interno del database, perché ci sarà un'opportuna funzione che trasformerà, sul front end, il link di iscrizione nell'opportuno Qr-code. Si implementa in qualsiasi linguaggio ma il più immediato, nel mondo web, resta sicuramente javascript (*articolo per approfondire: <https://www.html.it/articoli/creare-e-leggere-qr-code-in-javascript/>*)

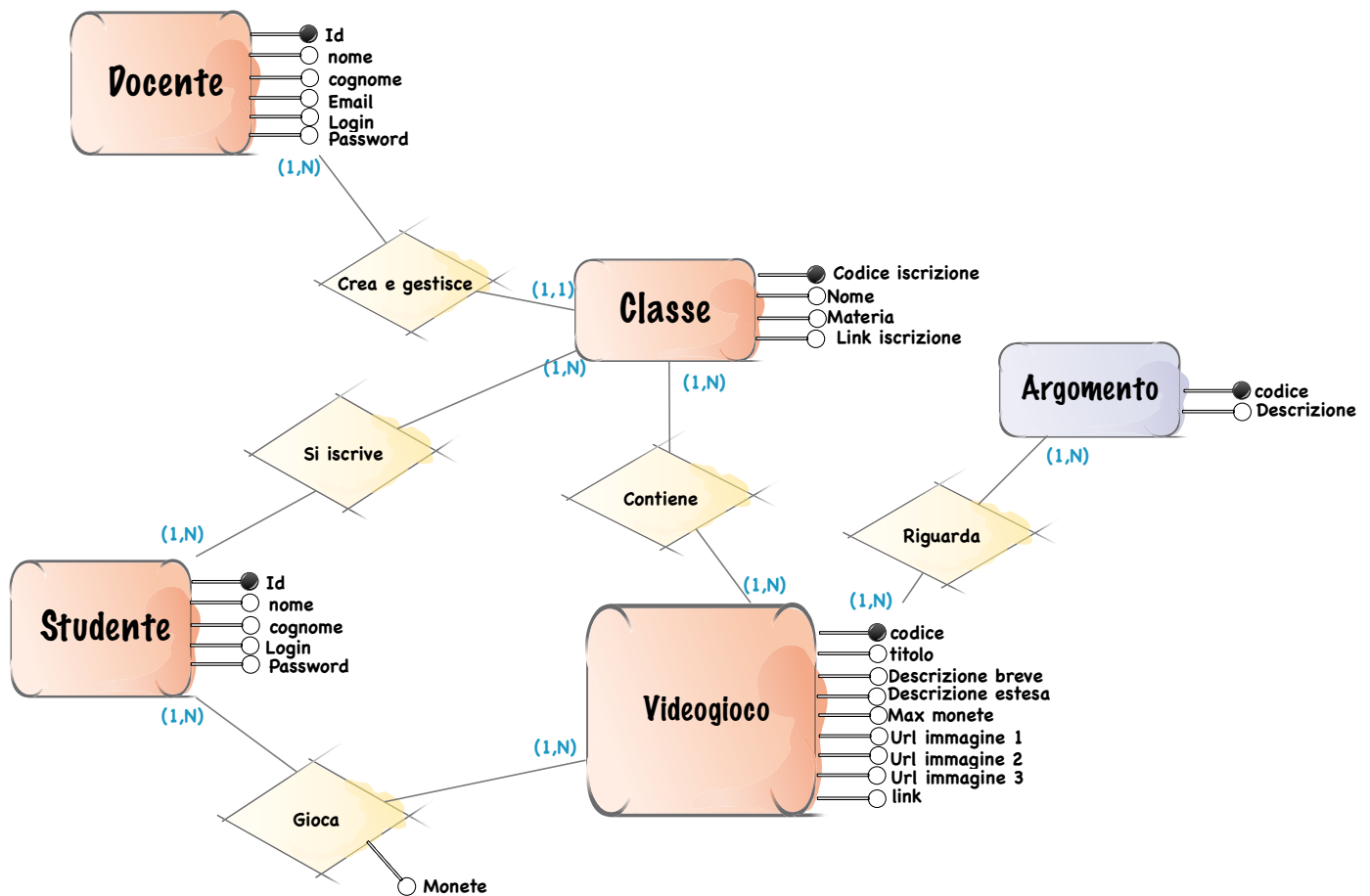
Di tutti gli attributi che sono semplici ed inequivocabili, l'attributo **immagine** per videogioco non diventa un'entità per il numero basso di immagini che possono essere riferite proprio al videogioco, al massimo tre. Essendo in ambiente web sappiamo che l'attributo immagine, di fatto, è l'url dell'immagine stessa, quindi una stringa di caratteri contenente l'indirizzo. Ma non è un errore inserirla come entità aggiungendo alcuni attributi come: *formato, risoluzione, didascalia*.

Volontariamente si può sorvolare sull'attributo '**quiz o attività**' che entra nel merito del contenuto del singolo videogioco che non è di pertinenza della traccia. Non si devono gestire i singoli videogiochi educativi ma la piattaforma. Ci affidiamo al link esterno ed eventualmente alla parte applicativa per gestire i giochi, i quiz e le attività. Non è utile, ai fini del nostro database, neanche indicare la tipologia sull'entità videogioco. Risulta chiaro che l'unica cosa di interesse, per la nostra base dati, sia la ricompensa in monete.

Passiamo a disegnare il modello concettuale con le sole entità e gli attributi:



Modello concettuale finale



Relazioni

Analizziamo le relazioni sulle entità individuate:

CREA E GESTISCE - DOCENTE CLASSE - Il docente crea N classi, una classe può essere creata da un solo docente. **Uno a molti**

SI ISCRIVE - STUDENTE CLASSE - Uno studente si può scrivere a più classi e una classe contiene più studenti. **Molti a molti**

CONTIENE - VIDEOGIOCO CLASSE - Un videogioco può essere associato a più classi, una classe contiene più videogiochi. **Molti a molti**

GIOCA - STUDENTE VIDEOGIOCO - Uno studente gioca a più videogiochi ed un videogioco viene giocato da più studenti. Giocando guadagna delle monete. **Molti a molti**

RIGUARDA - VIDEOGIOCO ARGOMENTO - Un videogioco può essere associato ad un certo argomento, ad un argomento vengono associati più videogiochi. **Molti a molti**

Osservazioni:

Anche se docente e studente hanno quasi gli stessi attributi, è da escludere una relazione di tipo ISA che potrebbe essere un generico UTENTE. Non cambia solo il ruolo all'interno della piattaforma ma anche la cardinalità della relazione con l'entità CLASSE. Non sarebbe un errore ma appare troppo evidente che nella trasformazione al logico poi andava applicato il collasso in basso per distinguere il DOCENTE dallo STUDENTE.

Quesito 2. Il relativo schema logico

Utilizziamo le regole di trasformazione per creare il modello logico.

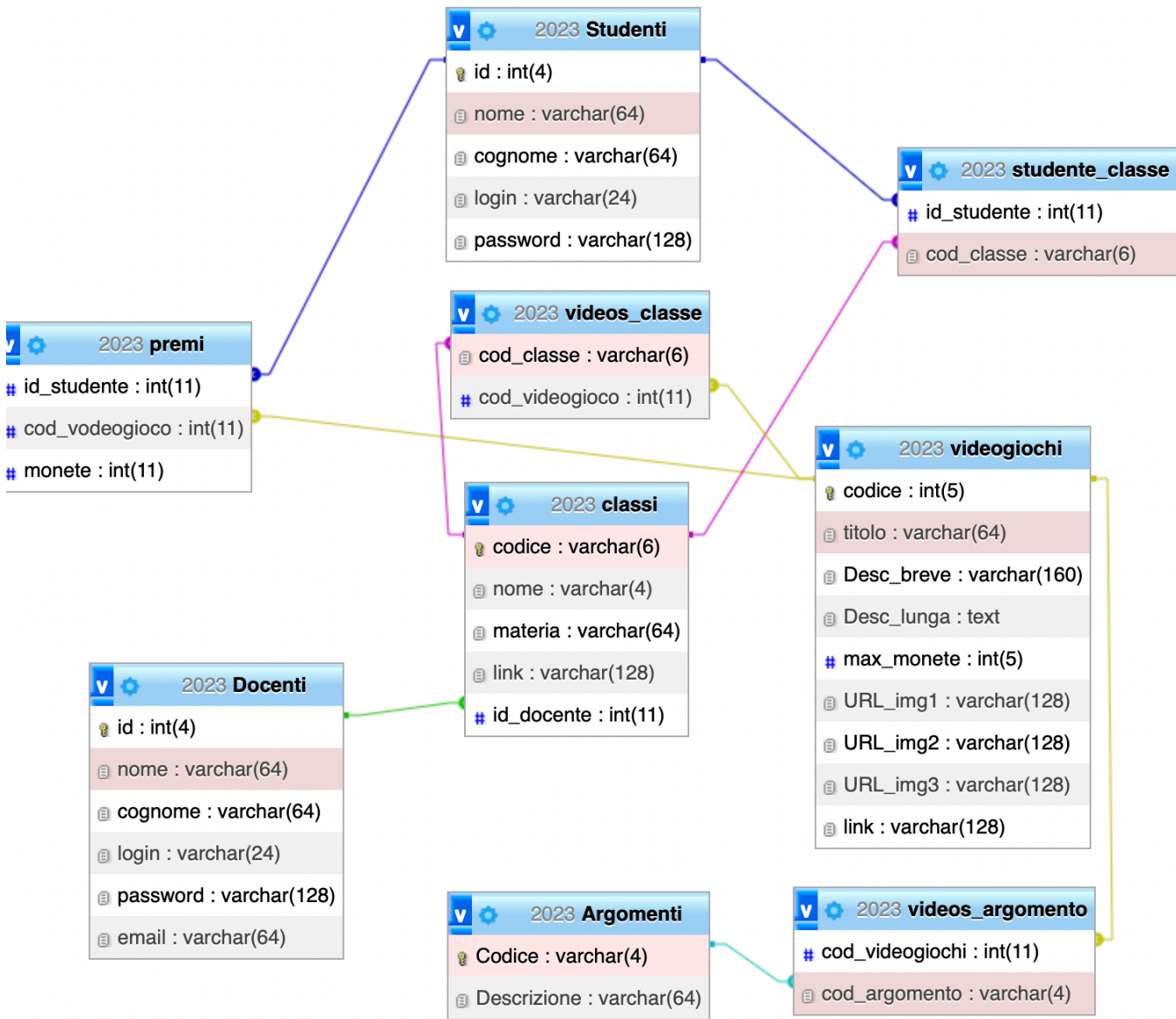
Traduzione molti a molti

Quattro relazioni il modello concettuale hanno una molteplicità molti a molti. Derivando il modello logico avremo quindi quattro tabelle in più. La tabella della relazione tra studenti e videogiochi prende la parola "**PREMI**" visto che contiene l'attributo **monete**.

Traduzione uno a molti

Le relazioni uno a molti si risolvono inserendo le chiavi esterne nelle tabelle a 'molteplicità' molti. Nel nostro caso all'interno della tabella classi la chiave esterna verso docenti.

Ed ecco il nostro modello logico:



Quesito 3. la definizione in linguaggio SQL di un sottoinsieme delle relazioni della base di dati in cui siano presenti alcune di quelle che contengono vincoli di integrità referenziale e/o vincoli di dominio, se esistenti;

Ecco il modello fisico, le CREATE_TABLE del MySQL:

```
CREATE TABLE `Argomenti` (  
  `Codice` varchar(4) NOT NULL,  
  `Descrizione` varchar(64) NOT NULL  
) ;
```

```
CREATE TABLE `classi` (  
  `codice` varchar(6) NOT NULL,  
  `nome` varchar(4) NOT NULL,  
  `materia` varchar(64) NOT NULL,  
  `link` varchar(128) NOT NULL,  
  `id_docente` int(11) NOT NULL  
) ;
```

```
CREATE TABLE `Docenti` (  
  `id` int(4) NOT NULL,  
  `nome` varchar(64) NOT NULL,  
  `cognome` varchar(64) NOT NULL,  
  `login` varchar(24) NOT NULL,  
  `password` varchar(128) NOT NULL,  
  `email` varchar(64) NOT NULL  
) ;
```

```
CREATE TABLE `premi` (  
  `id_studente` int(11) NOT NULL,  
  `cod_vodeogioco` int(11) NOT NULL,  
  `monete` int(11) NOT NULL  
) ;
```

```
CREATE TABLE `studente_classe` (  
  `id_studente` int(11) NOT NULL,  
  `cod_classe` varchar(6) NOT NULL  
) ;
```

```
CREATE TABLE `Studenti` (  
  `id` int(4) NOT NULL,  
  `nome` varchar(64) NOT NULL,  
  `cognome` varchar(64) NOT NULL,  
  `login` varchar(24) NOT NULL,  
  `password` varchar(128) NOT NULL  
) ;
```

```

CREATE TABLE `videogiochi` (
  `codice` int(5) NOT NULL,
  `titolo` varchar(64) NOT NULL,
  `Desc_breve` varchar(160) NOT NULL,
  `Desc_lunga` text NOT NULL,
  `max_monete` int(5) NOT NULL,
  `URL_img1` varchar(128) NOT NULL,
  `URL_img2` varchar(128) DEFAULT NULL,
  `URL_img3` varchar(128) DEFAULT NULL,
  `link` varchar(128) NOT NULL
) ;

```

```

CREATE TABLE `videos_argomento` (
  `cod_videogiochi` int(11) NOT NULL,
  `cod_argomento` varchar(4) NOT NULL
) ;

```

```

CREATE TABLE `videos_classe` (
  `cod_classe` varchar(6) NOT NULL,
  `cod_videogioco` int(11) NOT NULL
) ;

```

```
-- Vincoli per la tabella `classi`
```

```

ALTER TABLE `classi`
  ADD CONSTRAINT `docente` FOREIGN KEY (`id_docente`) REFERENCES `Docenti` (`id`);

```

```
--
-- Vincoli per la tabella `premi`
--
```

```

ALTER TABLE `premi`
  ADD CONSTRAINT `stud` FOREIGN KEY (`id_studente`) REFERENCES `Studenti` (`id`),
  ADD CONSTRAINT `videos` FOREIGN KEY (`cod_vodeogioco`) REFERENCES `videogiochi`
  (`codice`);

```

```
--
-- Vincoli per la tabella `studente_classe`
--
```

```

ALTER TABLE `studente_classe`
  ADD CONSTRAINT `studente_classe_ibfk_1` FOREIGN KEY (`cod_classe`) REFERENCES
  `classi` (`codice`),
  ADD CONSTRAINT `studente_classe_ibfk_2` FOREIGN KEY (`id_studente`) REFERENCES
  `Studenti` (`id`);

```

```
--
-- Vincoli per la tabella `videos_argomento`
--
```

```

ALTER TABLE `videos_argomento`
  ADD CONSTRAINT `arg` FOREIGN KEY (`cod_argomento`) REFERENCES `Argomenti` (`Codice`),
  ADD CONSTRAINT `video` FOREIGN KEY (`cod_videogiochi`) REFERENCES `videogiochi`
  (`codice`);

```

```
--
-- Vincoli per la tabella `videos_classe`
--
```

```

ALTER TABLE `videos_classe`
  ADD CONSTRAINT `videos_classe_ibfk_1` FOREIGN KEY (`cod_videogioco`) REFERENCES
  `videogiochi` (`codice`),
  ADD CONSTRAINT `videos_classe_ibfk_2` FOREIGN KEY (`cod_classe`) REFERENCES `classi`
  (`codice`);

```

Quesito 4 - le seguenti interrogazioni espresse in linguaggio SQL:

a) l'elenco in ordine alfabetico dei giochi classificati per uno specifico argomento;

```
SELECT titolo,cod_argomento
FROM videogiochi, videos_argomento
WHERE videogiochi.codice = videos_argomento.cod_videogiochi
AND videos_argomento.cod_argomento = 'MATE'
ORDER BY titolo;
```

b) la classifica degli studenti di una certa classe virtuale, in base alle monete raccolte per un certo gioco;

```
SELECT 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'
AND videogiochi.titolo = 'PITAGORIAMO'
ORDER BY monete DESC
```

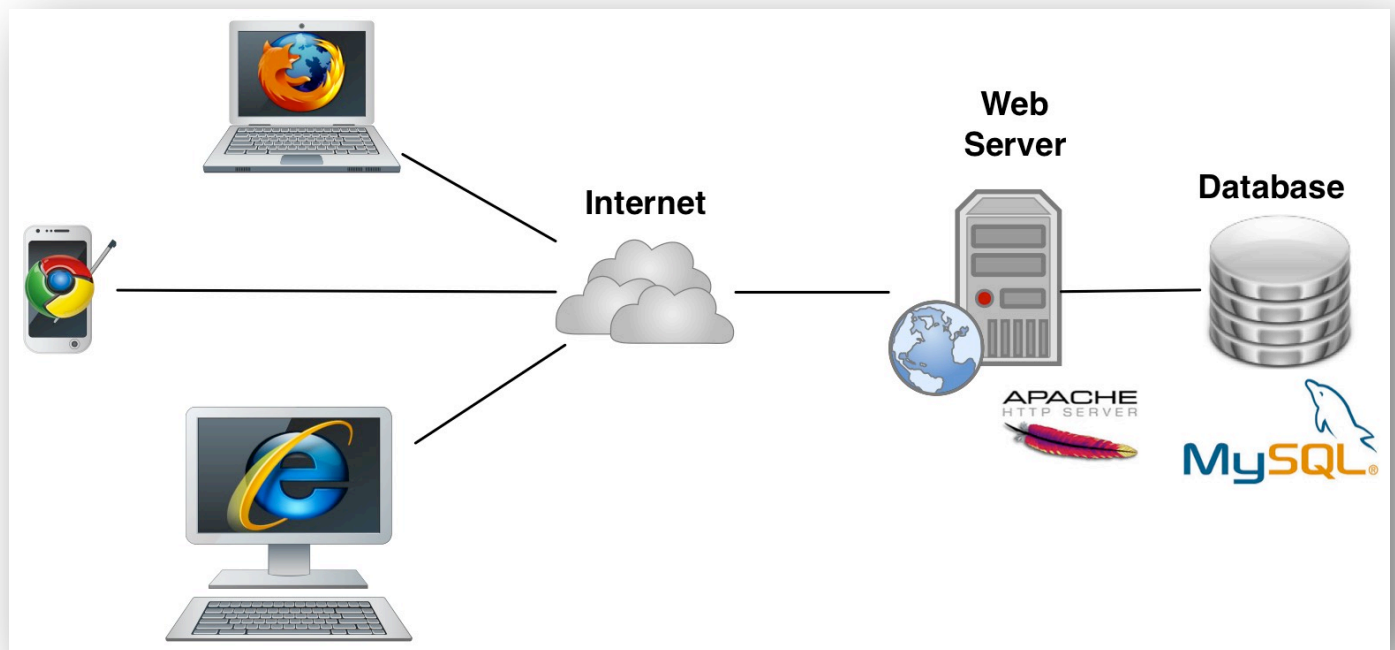
c) il numero di classi in cui è utilizzato ciascun videogioco del catalogo;

```
SELECT COUNT(*), videogiochi.titolo
FROM classi, videogiochi, videos_classe
WHERE videos_classe.cod_classe = classi.codice
AND videos_classe.cod_videogioco = videogiochi.codice
GROUP BY videos_classe.cod_videogioco
```

Quesito 5 - il progetto di massima della struttura dell'applicazione web per la gestione della realtà sopra presentata;

Per il software si ipotizza di utilizzare una architettura classica 3-tier:

- MySQL come Database
- Apache come Web Server
- Php come linguaggio lato Server
- HTML, CSS, Java Script per implementare le pagine Web.



Quesito 6 - una parte significativa dell'applicazione web che consente l'interazione con la base di dati, utilizzando appropriati linguaggi a scelta sia lato client che lato server.

Come parte significativa del codice si implementa in PHP la creazione della classe virtuale da parte del docente, quindi una insera all'interno della tabella CLASSI. Si presuppone che il docente abbia effettuato il login alla piattaforma e i suoi dati si trovino in sessione.

Form creata sul sito <https://bootsnipp.com/forms> in HTML e CSS tramite bootstrap.

Creazione Classe Virtuale

Nome	<input type="text" value="3B"/>
Materia	<input type="text" value="STORIA"/>
	<input type="button" value="Crea"/>

Codice della FORM

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

<form class="form-horizontal" action="insclasse2023.php" method="post" >
<fieldset>

<!-- Form Name -->
<legend>Creazione Classe Virtuale</legend>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-4 control-label" for="nome">Nome</label>
  <div class="col-md-4">
    <input id="nome" name="nome" type="text" placeholder="nome classe" class="form-
control input-md" required="">

  </div>
</div>

<!-- Text input-->
<div class="form-group">
  <label class="col-md-4 control-label" for="materia">Materia</label>
```

```

    <div class="col-md-4">
    <input id="materia" name="materia" type="text" placeholder="materia" class="form-
control input-md" required="">

    </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

```

<?php
session_start();

if(isset($_POST['nome'])) $nome = $_POST['nome'];
if(isset($_POST['materia'])) $materia = $_POST['materia'];
$id_docente = $_SESSION['id_utente'];

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "2023";

// Creare la connessione
$conn = new mysqli($servername, $username, $password, $dbname);
// Check della connessione
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
//creo codice classe di 6 caratteri random
$codice_classe = '';
for($i = 1; $i <= 6; $i++)
{
    $codice_ascii = rand(65,90); //range lettere maiuscole
    $codice_classe = $codice_classe . chr($codice_ascii);
}
//creazione link con il codice classe
$link = "http://piattaformaesame2023/" . $codice_classe;

$sql = "INSERT INTO classi(codice, nome, materia, link, id_docente) ".
" VALUES ('" . $codice_classe . "', '" . $nome . "', '" . $materia . "', '" . $link . "', ".
$id_docente . "));";

if ($conn->query($sql) === TRUE) {
    echo "<br>Classe inserita correttamente! " ;
} else {
    echo "Errore durante la creazione: " . $sql . "<br>" . $conn->error ."Si prega di
ritentare.";
    //Per scongiurare il rischio remoto, così remoto da rendere inutile un controllo
    //ma in caso di chiave primaria duplicata si richiede di reinserire la classe
}
$conn->close();
?>

```