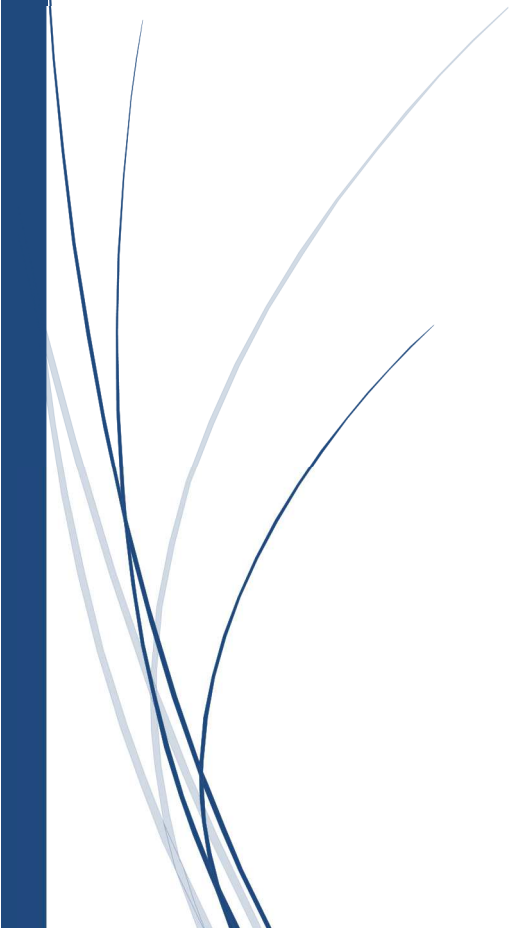




23/07/2019

CORSI ONLINE

Progetto modalità tradizionale



Nicole Vignone
Alessandro Magno

PROGETTAZIONE

SCHEMA ER

SCHEMA RELAZIONALE

CODICE SQL

VINCOLI

CREAZIONE TABELLE

RIEMPIMENTO TABELLE

INTERROGAZIONI

FUNZIONI E TRIGGER

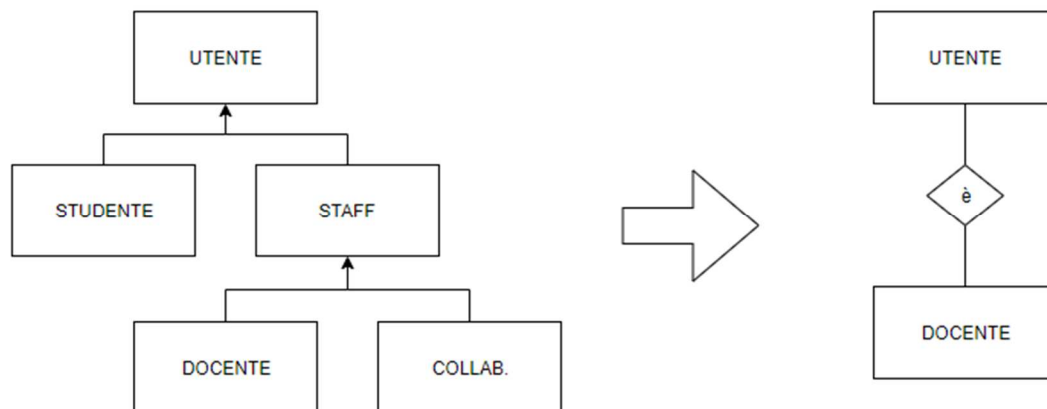
ELIMINAZIONE GERARCHIE

L'entità ristrutturate sono state:

- Utente
- Forum
- Assignment

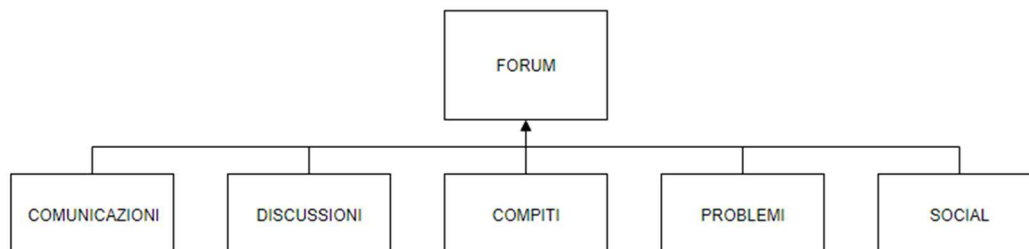
UTENTE

Gli studenti e i collaboratori vengono contraddistinti dall'attributo ruolo. L'entità docente è stata lasciata indicata, poiché ha più relazioni specifiche, che lasciate indicate con l'entità utente, avrebbero perso di significato.

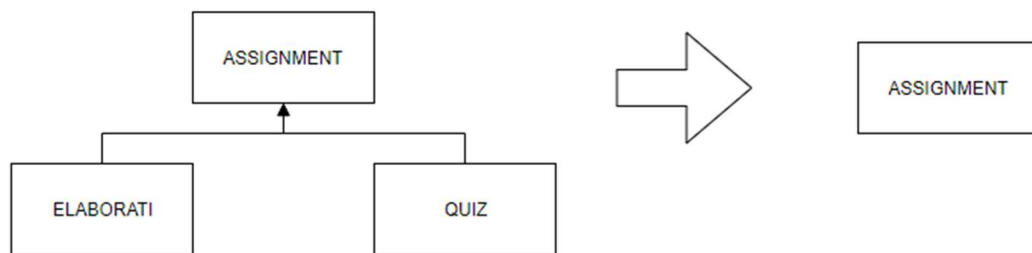


FORUM

I vari forum vengono riconosciuti tramite l'attributo tipo (1. Comunicazioni, 2. Discussioni...). Pertanto, l'unica entità lasciata è Forum.



ASSIGNMENT



Le due entità figlie vengono eliminate e riconosciute tramite l'attributo Tipo dell'entità padre Assignment.

NB: Tutti gli attributi omissi in queste rappresentazioni, sono ben presenti sullo schema completo

ELIMINAZIONE ATTRIBUTI MULTI-VALORE

L'attributo multi-valore Social è stato trasformato in un'entità a parte, con la chiave primaria Nome.

Gli attributi multi-valore prerequisiti, sommario e letture, sono diventati attributi mono valore, che in fase di traduzione verranno indicati come testi (tipo text o varchar in SQL).

SCHEMA RELAZIONALE

Chiave Primaria

Chiave Alternativa

SCHEMA UNIVERSITA

UNIVERSITA(Nome, Descrizione, Logo, Link, Indirizzo)

DOMINI

Dom(Nome, Descrizione, Logo, Link, Indirizzo) = String

RELAZIONE

String x String x String x String x String

SCHEMA CORSO

CORSO(Id, Università^{UNIVERSITA}, Titolo, Descrizione, Sommario, Prerequisiti, Letture_o, Modalità, Soglia, Lingua)

DOMINI

Dom(Id, Università, Titolo, Descrizione, Sommario, Prerequisiti, Letture, Lingua) = String

Dom(Modalità) = Char

Dom(Soglia) = Int

RELAZIONE

String x String x String x String x String x String x String x Char x Int x String

SCHEMA CATEGORIA

CATEGORIA(Nome)

DOMINI

Dom(Nome) = String

RELAZIONE

String

SCHEMA APPARTIENE

APPARTIENE(Id^{CORSO}, Nome^{CATEGORIA})

DOMINI

Dom(Id, Nome) = String

RELAZIONE

String x String

SCHEMA MATERIALE

MATERIALE(Id, *Corso*^{CORSO}, *Titolo*, *Durata*, *Sottotitolo*₀, *Lingua*, *Descrizione*, *Tipo*, *Argomento*₀)

DOMINI

Dom(Id, Corso, Titolo, Lingua, Descrizione, Argomento) = String

Dom(Durata) = Decimal

Dom(Sottotitoli) = Bool

Dom(Tipo) = Char

RELAZIONE

String x String x String x Decimal x Bool x String x String x Char x String

SCHEMA EDIZIONE

EDIZIONE(NumeroEd, Corso^{CORSO}, *Durata*, *DataIn*₀, *Ore*)

DOMINI

Dom(Corso) = String

Dom(NumeroEd, Ore) = Int

Dom(Durata) = Decimal

Dom(DataIn) = Data

RELAZIONE

Int x String x Decimal x Data x Int

SCHEMA ATTIVITA

ATTIVITA(Id, Corso^{EDIZIONE}, Edizione^{EDIZIONE}, DataIn, DataF)

DOMINI

Dom(Id, Corso) = String

Dom(DataIn, DataF) = Data

Dom(Edizione) = Int

RELAZIONE

String x String x Int x Data x Data

SCHEMA UTENTE

UTENTE(Email, Nome, Cognome, Ruolo, WL)

DOMINI

Dom(Email, Nome, Cognome, Ruolo, WL) = String

RELAZIONE

String x String x String x String x String

SCHEMA DOCENTE

DOCENTE(Email^{UTENTE}, Universita^{UNIVERSITA}, Affiliazione, Ruolo, Foto, CV)

DOMINI

Dom(Email, Universita, Affiliazione, Ruolo, Foto, CV) = String

RELAZIONE

String x String x String x String x String x String

SCHEMA FORUM

FORUM(Id^{CORSO}, Tipo, Chiuso)

DOMINI

Dom(Id) = String

Dom(Tipo) = Int

Dom(Chiuso) = Bool

RELAZIONE

String x Int x Bool

SCHEMA THREAD

THREAD(Id, Utente^{UTENTE}, Corso^{FORUM}, Forum^{FORUM}, Chiuso)

DOMINI

Dom(Id, Utente, Corso) = String

Dom(Forum) = Int

Dom(Chiuso) = Bool

RELAZIONE

String x String x String x Int x Bool

SCHEMA MESSAGGI

MESSAGGI(Id, Thread^{THREAD}, Testo, Risposta^{MESSAGGI}_o, Utente^{UTENTE})

DOMINI

Dom(Id, Thread, Testo, Risposta, Utente) = String

RELAZIONE

String x String x String x String x String

SCHEMA PROFILO

PROFILO(Utente^{UTENTE}, Genere_o, Foto_o, Descrizione_o, Località_o, DataN_o, WebP_o)

DOMINI

Dom(Utente, Foto, Descrizione, Località, WebP) = String

Dom(Genere) = Char

Dom(DataN) = Data

RELAZIONE

String x Char x String x String x String x Data x String

SCHEMA LINK

LINK(Social^{SOCIAL}, Utente^{PROFILO})

DOMINI

Dom(Social, Utente) = String

RELAZIONE

String x String

SOCIAL(**Nome**)

DOMINI

Dom(Nome) = String

RELAZIONE

String

SCHEMA ASSIGNMENT

ASSIGNMENT(**Id**, DataIn, DataF, Consegna, HDLo, Tipo, Soglia, Peer)

DOMINI

Dom(Id) = String

Dom(DataIn, DataF, Consegna, HDL) = Data

Dom(Tipo) = Char

Dom(Soglia) = Int

Dom(Peer) = Bool

String x Data x Data x Data x Data x Char x Int x Bool

SCHEMA DOMANDE

DOMANDE(**Quiz**^{ASSIGNMENT}, **IdDom**, Testo)

DOMINI

Dom(Quiz, IdDom, Testo) = String

RELAZIONE

String x String x String

SCHEMA RISPOSTE

RISPOSTE(**Quiz**^{DOMANDE}, **Domanda**^{DOMANDE}, Testo, Punti)

DOMINI

Dom(Quiz, Domanda, Testo) = String

Dom(Punti) = Int

RELAZIONE

String x String x String x Int

SCHEMA CONSEGNA

CONSEGNA(**DataC**, **Assignment**^{ASSIGNMENT}, **Utente**^{UTENTE}, Voto, Pen)

DOMINI

Dom(Assignment, Utente) = String

Dom(Voto) = Decimal

Dom(DataC) = Data

Dom(Pen) = Bool

RELAZIONE

Data x String x String x Decimal x Bool

SCHEMA SCELTA

SCELTA(Utente^{UTENTE}, Risposta^{RISPOSTA}, Domanda^{RISPOSTA}, Quiz^{RISPOSTA}, DataS, Penalizzazione)

DOMINI

Dom(Utente, Risposta, Domanda, Quiz) = String

Dom(DataS) = Data

Dom(Penalizzazione) = Bool

RELAZIONE

String x String x String x String x Data x Bool

SCHEMA PEER

PEER(Docente^{DOCENTE}, Studente^{UTENTE}, Elaborato^{CONSEGNA}, Esecutore^{CONSEGNA}, DataEs^{CONSEGNA}, dataAss, Voto, Peso, DataV)

DOMINI

Dom(Docente, Studente, Elaborato, Esecutore) = String

Dom(DataEs, DataAss, DataV) = Data

Dom(Voto) = Decimal

Dom(Peso) = Decimal

RELAZIONE

String x String x String x String x Data x Data x Decimal x Decimal x Data

SCHEMA INSEGNA

INSEGNA(Docente^{DOCENTE}, Corso^{CORSO})

DOMINI

Dom(Docente, Corso) = String

RELAZIONE

String x String

SCHEMA VISIONATO

VISIONATO(Video^{MATERIALE}, Utente^{UTENTE})

DOMINI

Dom(Video, Utente) = String

RELAZIONE

String x String

SCHEMA ISCRITTO

ISCRITTO(Utente^{UTENTE}, Corso^{EDIZIONE}, Edizione^{EDIZIONE}, Superato);

DOMINI

Dom(Utente, Corso) = String

Dom(Edizione) = Int

Dom(Superato) = Bool

RELAZIONE

String x String x Int x Bool

SCHEMA HASUPERATO

HASUPERATO(Utente^{UTENTE}, Esercizio^{ASSIGNMENT}, Voto)

DOMINI

Dom(Utente, Esercizio) = String

Dom(Voto) = Decimal

RELAZIONE

String x String x Decimal

SCHEMA PREVEDE

PREVEDE(Materiale^{MATERIALE}, Attivita^{ATTIVITA})

DOMINI

Dom(Materiale, Attivita) = String

RELAZIONE

String x String

Gli attributi che nella realtà sono immagini, nel database vengono salvate come String, in quanto viene indicato il loro percorso.

VINCOLI

- Nelle relazioni Corso, Materiale, Attività, Assignment, Messaggi, Thread, la chiave primaria è l'attributo Id, aggiunto ad hoc.
- Nelle relazioni Università, la chiave primaria è il Nome della stessa. Idem per le relazioni Categoria e Social.
- Nelle relazioni Peer, Appartiene, HaSuperato, Link, Iscritto, le chiavi primarie sono composte dalle chiavi primarie delle relazioni partecipanti.
- Nelle relazioni Domande, Risposte, Forum, Scelta, Consegna, Edizione, le chiavi primarie sono composte da attributi della stessa e chiavi primarie provenienti da relazioni ad essa associate.
- Nella relazione Peer, la data di assegnazione dell'elaborato da parte del docente ad uno degli studenti deve essere successiva alla data di consegna prevista per quell'assignment. Inoltre, chiaramente, l'elaborato in questione deve essere stato consegnato per tempo, ovvero tra la data di consegna e quella di HardDeadline (può consegnarlo anche prima della data di consegna).
- Chi consegna in ritardo, non può partecipare alla peer evaluation.
- Chi consegna in ritardo la peer evaluation, prende una penalizzazione sul suo voto, pari al 20% del voto (28 -> 22.4)
- Se il corso si presenta in modalità Self-Study, non vengono create attività. Il materiale viene prelevato direttamente dalla tabella Materiale e reso disponibile seguendo le direttive dell'attributo Argomento della relazione Materiale.
Le dipendenze tra gli argomenti sono specificate nell'attributo prerequisiti.
Per questo tipo di corsi, vengono dati solo assignment di tipo quiz. Non sussiste il vincolo di HDL, tantomeno una penalizzazione.
Rimane solo il fatto che fintanto che non vengono completati tali quiz non si può procedere con le lezioni successive.
- Se gli assignment vengono consegnati in ritardo, ovvero dopo la HDL, viene applicata d'ufficio una penalizzazione del 40% sul voto inizialmente preso per quell'assignment (Voto preso: 28 -> 16.8).
Se l'assignment è di tipo quiz, e alcune risposte risultano mancanti dopo la scadenza della HDL, non possono essere più completate. Nel calcolo del voto verranno conteggiate solo quelle presenti nella relazione Scelta. Se il punteggio risulta sufficiente, l'assignment è superato.
- Il voto totale di un assignment di tipo esercizio è dato dalla media tra l'attributo voto della tabella Consegna e l'attributo voto della tabella Peer (in questo caso si tiene conto di quanto pesa nella media). Se il voto è superiore alla soglia impostata per l'assignment, viene inserito nella relazione haSuperato, con il suo voto.

Il voto totale di un assignment di tipo quiz è dato dalla somma dei punti delle risposte date. Se risulta uguale o superiore alla soglia imposta viene inserito nella relazione haSuperato. Per sapere se il corso è stato superato, si controlla dapprima che nella relazione haSuperato ci siano tutti gli assignment previsti per quel corso. In caso favorevole, si esegue una media aritmetica tra i voti. Se uguale o superiore alla soglia imposta dal corso, ed inoltre sono stati visionati tutti i video previsti (si controlla che nella relazione visionato, per il singolo utente, ci siano un numero di tuple pari a quelle dei video), tale corso risulta essere superato. In caso contrario, l'utente è libero di riscrivere ad una nuova edizione, se prevista, di tale corso.

- Nella relazione consegna possono essere inseriti solo assignment di tipo esercizio interattivo. Nella relazione domande possono essere inseriti solo gli assignment di tipo quiz.
- Se l'HDL non è specificata ci si trova davanti ad un corso in modalità self-study.

CREAZIONE TABELLE

- Foglio 1 : CreateTablees.sql

INSERIMENTO

- Foglio 2: Universita_Utente_Docente.sql
- Foglio 3: Categoria_Corso_Appartiene.sql
- Foglio 4: Materiale.sql
- Foglio 5: Insegna_Edizione_Attivita.sql
- Foglio 6: Assignment_Peer_Consegna_Iscritto.sql
- Foglio 7: Social_Profilo_Forum_Link.sql
- Foglio 8: Prevede.sql
- Foglio 9: Visionato.sql

INTERROGAZIONI

- Foglio 10: Query.sql

FUNZIONI E TRIGGER

- Foglio 11: View&Funzioni.sql
- Foglio 12: Trigger.sql

Per le funzioni sono state create apposite view, per una fruizione degli attributi necessari più veloce

FUNZIONI

- **CalcolaVoto**: per ogni singolo utente, calcola il voto conseguito dopo la consegna "normale" di ogni singolo elaborato. Se la consegna fosse avvenuta in ritardo, applica la penalizzazione dovuta.
- **PenalizzazionePeer**: calcola il voto finale della valutazione peer. Se ci sono state consegne in ritardo, applica la penalizzazione.
- **InsertNotPeer**: inserisci gli assignment di tipo esercizio, che non partecipano alla valutazione peer. Anche in questo caso, i voti sono comprensivi di penalizzazione, se dovuta.
- **SuperamentoQuiz**: calcola il punteggio conseguito nei quiz. Se superiore ad una certa soglia imposta dall'assignment, viene inserito nella tabella HaSuperato.

- **CalcolaMedia**: calcola il voto medio, dato il voto preso nella valutazione tra studenti e quello dato dal docente. Se superiore alla soglia imposta dall'assignment, viene inserito nella relazione HaSuperato.
- **CorsiSuperati**: controlla che la media conseguita sia superiore alla soglia imposta dal corso e che siano stati visionati tutti i video. In caso favorevole setta a true l'attributo superato della relazione Iscritto.

Le prime quattro funzioni possono essere richiamate alla fine di ogni settimana di attività, o alla fine del corso. Le ultime due possono essere richiamate solo alla fine del corso. Se venissero chiamate prima, ovviamente i conteggi non risulterebbero essere corretti.

TRIGGER

- **CheckConsegna**: Controlla che la data in cui viene assegnato, da parte di un docente, un assignment di tipo esercizio, per la valutazione tra studenti, sia posteriore alla data di consegna prevista per quell'assignment
- **CheckEsercizio**: Controlla che gli assignment di tipo esercizio vengano inseriti solo nella tabella consegna, e non anche nella tabella Domande, relativa ai quiz. Viceversa, se l'assignment è di tipo quiz non deve finire nella tabella consegna
- **CheckHDL**: controlla che, se definita, la Hard Dead Line, sia posteriore alla data di consegna imposta per un determinato assignment
- **CheckIscrizione**: controlla che gli utenti si iscrivano ad edizioni di corsi per cui è già stata definita la data di inizio, e che non superi i 7 giorni da quest'ultima
- **CheckPeer**: controlla che gli assignment che hanno l'attributo peer a false, non vengano inseriti e usati per la valutazione tra studenti
- **CheckUtenteConsegna**: controlla che gli utenti iscritti ad un determinato corso, consegnino solo gli assignment inerenti al loro corso
- **CheckVideo**: controlla che gli utenti iscritti ad un determinato corso, possano guardare solo i video inerenti al loro corso
- **DoppiaIscrizione**: controlla che un utente che ha già superato l'edizione di un determinato corso, non si iscriva una seconda volta ad una nuova edizione dello stesso corso.