# **DIZIONARIO DELLE CLASSI**

|  |  |  |
| --- | --- | --- |
| **NOME** | **DESCRIZIONE** | **ATTRIBUTI** |
| **UTENTE** | Contiene tutti gli utenti standard che entrano ed interagiscono con gruppi ed altri utenti. | **Nickname**(Varchar(30)):  L’identificativo con il quale un utente si registra.  **Password** (Varchar(20))  parola chiave con la quale un utente entra nel suo profilo.  **Numero\_gruppi(integer)**  Numero di gruppi ai quali l’utente è iscritto  **idUtente(varchar(10))**  identificativo univoco associato casualmente ad ogni utente(vincolo reminder: ogni id dovrà essere diverso dall’altro)  **bio(varchar(200))**  descrizione del profilo. |
| **GRUPPO** | Contiene tutti i gruppi che sono stati creati dagli utenti e sui quali interagiscono postando contenuti. | **idGruppo(varchar(10))**  Identificativo univoco di ogni gruppo.  **Nome\_gruppo(varchar(30))**  Nome scelto dal creatore.  **Descrizione gruppo (varchar(200))**  Testo di informazioni generali sul gruppo.  **Data\_creazione(date)**  Data nella quale è stato creato il gruppo |
| **POST** | Contiene tutti i post caricati dagli utenti sul social | **idPost(varchar(10))**  identificativo univoco di ogni post  **idUtente(varchar(10))**  riferimento esterno all’utente che ha effettuato il caricamento.  **idGruppo(varchar(10))**  riferimento esterno al gruppo sul quale il post verrà caricato.  **Descrizione(varchar(30))**  Testo semplice scritto dall’utente al caricamento del post.  **Numero\_like(integer)**  Numero di like ricevuti dal post, derivato dall’associazione con recensione.  **Numero\_commenti(integer)**  Numero di commenti ricevuti dal post, derivato dall’associazione con recensione.  **Data\_pubblicazione(date)**  Data nella quale è stato caricato il post sul gruppo.  **Orario\_pubblicazione(time)**  Orario di pubblicazione del post |
| **RECENSIONE** | Contiene tutte le interazioni con un post che siano like o commenti | **idUtente(varchar(10))**  riferimento all’utente che ha lasciato l’interazione con il post.  **idPost(varchar(10))**  riferimento al post sul quale viee lasciata la recensione.  **Data\_recensione(date)**  Data nella quale è stata lasciata la recensione  **Orario\_recensione(time)**  Orario nel quale è stata lasciata la recensione  **Tipo(boolean)**  Assume 2 valori che determinano se la recensione sia un like, ciò porterà ad un incremento del numero di like del post riferito e renderà NULL l’attributo descrizione o un commento, che farà la stessa cosa con i commenti e aggiungerà un testo all’attributo descrizione |
| **Notifica** | Contiene tutte le informazioni inerenti alle notifiche | **IdUtente(varchar(10))**  Riferimento All’Utente che riceve la notifica  **IdPost(varchar(10))**  Riferimento al contenuto creato il quale di fatto scaturisce la notifica  **IdGruppo(varchar(10))**  Riferimento al Gruppo da cui proviene la notifica  **Data\_Notifica(date)**  Data Di Ricevimento Della Notifica  (ovvero la data di creazione del contenuto)  **Orario\_Notifica(time)**  Orario Di Ricevimento Notifica (ovvero l’orario di creazione del contenuto)  **Descrizione\_Notifica(varchar(200))**  Contenuto Testuale Della Notifica |
| **Tag** | Contiene tutti i Tag | **IdTag(varchar(10))**  **Testo(varchar(20))**  **IdGruppo(varchar(10))** |
| **CREA** | Classe di associazione tra l’utente che crea un gruppo e il gruppo stesso | **idUtente(varchar(10))**  id dell’utente esterno che compie l’atto di creare il gruppo  **idGruppo(varchar(10))**  id del gruppo che viene creato |
|  |  |  |

# **DIZIONARIO DELLE ASSOCIAZIONI**

|  |  |  |
| --- | --- | --- |
| **NOME** | **DESCRIZIONE** | **CLASSI COINVOLTE** |
| **PARTECIPA** | Descrive l'associazione di partecipazione con una classe Gruppo. | **Utente[0...\*]:** indica gli utenti che si registrano all'interno di un gruppo.  **Gruppo[0...\*]:** indica i gruppi del social ai quali gli utenti si iscrivono per interagire con altri utenti. |
| **INVIA** | Descrive il processo per il quale un gruppo invia una notifica relativa a un post. | **Gruppo[1]:** Indica la classe gruppo che si occuperà di inviare la notifica.  **Notifica[0...\*]:** indica la classe notifica che verrà inviata a un utente una volta che qualcuno pubblica un post in un gruppo a cui è iscritto. |
| **INVIATA\_A** | Associazione che descrive il legame tra la notifica che viene inviata all'utente iscritto ad un determinato gruppo e l'utente che riveve suddetta notifica. | **Utente[1...\*]:** partecipazione che descrive il concetto che una notifica può arrivare a nessun utente [0] nel caso non ci siano altri partecipanti in un gruppo o nel caso ce n'è siano di più viene inviata a più utenti[\*].  **Notifica[0...\*]:** la classe notifica nell'associazione ha la parte di rappresentare le notifiche inviate ad uno o più utenti che ricevono 0 o più notifiche. |
| **ACCETTA** | Associazione ricorsiva che lega due utenti e specifica il gesto di accettare o meno all'interno di un gruppo un utente che fa richiesta, associazione possibile solo nel caso che da una delle due parti vi sia un utente creatore. | **Utente[1]:** utente obbligatoriamente creatore di u gruppo che compie il gesto di accettare una richiesta di entrare in un gruppo.  **Utente[0...\*]:** nessuno o più utenti che richiedono allo stesso creatore di entrare in un gruppo. |
| **EFFETTUA** | Associazione che lega una recensione all'utente che la effettua su un post. | **Utente[1]:** L'utente univoco che scrive un commento o lascia un like ad una pubblicazione.  **Recensione[0...\*]:** zero o più recensioni pubblicate da uno stesso utente per uno o più post diversi. |
| **RELATIVA\_A** | Associazione che descrive il legame tra una o più recensioni lasciate dagli utenti su un post. | **Recensione[0...\*]:** classe recensione che indica il concetto che 0 o molte recensioni possono appartenere ad un unico post.  **Post[1]:** classe post unico al quale vengono associate più recensioni da diversi utenti. |
| **PUBBLICA** | Associazione che descrive l'azione di un utente di pubblicazione di un post in un gruppo al quale iscritto. | **Utente[1]:**classe che indica che un post è pubblicato da uno e un solo utente.  **Post[0...\*]:**classe che indica che un utente può pubblicare nessuno[0] o più post[\*] |
| **RELATIVO\_A** | Associazione che descrive il legame tra una o più pubblicazioni (post) relative ad un gruppo. | **Post[0...\*]:**classe che specifica il tipo di contenuto pubblicabile e le interazioni ottenibili su di esso.La molteplicità relativa alla classe,vuole indicare la possibilità di esserci **[0]** post in un gruppo o, molti **[\*]**.  **Gruppo[1]:** La classe gruppo specifica l'univoco**[1]** gruppo a cui puo appartenere un post. |
|  |  |  |
|  |  |  |

**DIZIONARIO DEI VINCOLI**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome vincolo | TIPO | Descrizione | Identificativo |
| Id\_utente univoco |  | L’id utente dovrà essere univoco per ogni utente e non potranno mai esserci duplicati. | UTENTE\_UK1 |
| UTENTE accetta solo se creatore |  | Un’entità utente può accettare un altro utente se e solo se è creatore nel suddetto gruppo della richiesta. | CREATORE\_CHECK |
| Data recensione maggiore data Post |  | La data di pubblicazione di una recensione deve essere più in avanti col tempo di quella del post a cui fa riferimento. | RECENSIONE\_DATA\_CHECK |
| Orario Recensione maggiore orario post |  | L’orario di una recensione effettuata nello stesso giorno di un post deve essere sempre maggiore dell’orario di pubblicazione del post. | RECENSIONE\_ORARIO\_CHECK |
| Enumerazione Tag Gruppo |  | I tag di un gruppo possono essere solo : Fantasy,Anime,Cartoni animati,Sport,Film,Videogiochi anche contemporaneamente. | GROUP\_TAG\_ENUMERATION |
| Id\_gruppo univoco |  | L’id di un gruppo dovrà essere univoco per ogni utente e non potranno mai esserci duplicati. | UTENTE\_UK1 |
| Tipo recensione può assumere 2 valori |  | L’attributo tipo di recensione può assumere o valore ‘like’ o valore ‘commento’ | RECENSIONE\_TIPO\_CHECK1 |
| Descrizione NULL se recensione tipo ‘like’ |  | Se il tipo di una recensione corrisponde a ‘like’ la sua descrizione dovra essere NULL | RECENSIONE\_TIPO\_CHECK2 |
| Numero di POST in un gruppo non negativo |  | Il numero di post pubblicati in un gruppo non può essere negativo. | GROUP\_CHECK2 |
| Numero di Gruppi non negativo |  | Il numero di gruppi ai quali un utente appartiene non può essere negativo. | UTENTE\_CHECK1 |
| Numero di POST non negativo |  | Il numero di post pubblicati da un utente non può essere negativo. | UTENTE\_CHECK2 |
| Creatore può assumere solo 2 valori |  | L’attributo creatore può assumere solo valore di si o no | CREATORE\_CHECK1 |
| Data notifica maggiore data Post |  | La data di pubblicazione di una notifica deve essere più in avanti col tempo di quella del post a cui fa riferimento. | NOTIFICA\_DATA\_CHECK |
| Orario notifica maggiore orario post |  | L’orario di una notifica effettuata nello stesso giorno di un post deve essere sempre maggiore dell’orario di pubblicazione del post. | NOTIFICA\_ORARIO\_CHECK |
| L’id\_utente della notifica stesso dell’utente |  | L’id utente della notifica deve referenziato a quello dell’utente al quale sarà inviato. | NOTIFICA\_CHECK\_UTENTE |
| Numero\_like non negativo |  | Il numero dei like di un post non deve essere negativo. | POST\_CHECK1 |
| Numero\_commenti non negativo |  | Il numero dei commenti di un post non deve essere negativo. | POST\_CHECK2 |
| Aumento like solo se recensione tipo like |  | Il numero dei like in un post aumenta solo se la recensione è di tipo like. | RECENSIONE\_CHECK1 |
| Aumento commenti solo se recensione tipo commenti |  | Il numero dei commenti in un post aumenta solo se la recensione è di tipo commento. | RECENSIONE\_CHECK2 |
| Enumerazione tipo notifica | Vincolo Intrarelazionale | Una notifica potrà assumere solo uno tra i 3 valori della sua enumerazione cioè: di Accettazione, Pubblicazione, Richiesta |  |
|  |  |  |  |

**SCHEMA FISICO**

CREAZIONE DELLE TABELLE

CREATE TABLE notifica (

idUtente varchar(10),

idPost varchar(10),

idGruppo varchar(10),

data\_notifica date,

orario\_notifica timestamp,

CONSTRAINT pk\_notifica PRIMARY KEY (idPost, idGruppo)

);

CREATE TABLE Utente (

nickname varchar(20) NOT NULL,

password varchar(8)NOT NULL ,

idUtente varchar(10) NOT NULL PRIMARY KEY,

numero\_gruppi integer,

bio varchar(100),

creatore BOOLEAN

);

CREATE TABLE recensione (

idUtente VARCHAR(10),

orario TIMESTAMP,

data\_recensione DATE,

idPost VARCHAR(10),

descrizione VARCHAR(300),

tipo VARCHAR(10),

CONSTRAINT pk\_recensione PRIMARY KEY (idUtente, orario, data\_recensione)

CONSTRAINT fk\_recensione\_utente FOREIGN KEY (idUtente)

REFERENCES utente(idUtente);

CONSTRAINT fk\_recensione\_post FOREIGN KEY (idPost)

REFERENCES post(idPost);

);

CREATE TABLE post (

idPost SERIAL PRIMARY KEY,

idUtente VARCHAR(10),

idGruppo VARCHAR(10),

descrizione VARCHAR(300),

numero\_like INTEGER,

numero\_commenti INTEGER,

data\_pubblicazione DATE,

orario\_pubblicazione TIMESTAMP,

CONSTRAINT fk\_utente FOREIGN KEY (idUtente) REFERENCES Utente(idUtente),

CONSTRAINT fk\_gruppo FOREIGN KEY (idGruppo) REFERENCES Gruppo(idGruppo)

);

CREATE TABLE notifica(

idPost VARCHAR(10),

idUtente VARCHAR(10),

idGruppo VARCHAR(10),

data\_notifica DATE,

orario\_notifica TIMESTAMP,

descrizione\_notifica VARCHAR(20)

CONSTRAINT

foreign\_key\_utente FOREIGN KEY(idutente) references utente(idutente)

CONSTRAINT

foreign\_key\_post FOREIGN KEY(idpost) references post(idpost);

CONSTRAINT

foreign\_key\_gruppo FOREIGN KEY(idgruppo) references gruppo(idgruppo);

);

CREATE TABLE richiesta (

idRichiedente VARCHAR(10),

idGruppo VARCHAR(10),

accettata BOOLEAN,

orario\_richiesta TIMESTAMP,

data\_richiesta DATE,

CONSTRAINT fk\_richiesta\_utente FOREIGN KEY (idRichiedente) REFERENCES utente(idUtente),

CONSTRAINT fk\_richiesta\_gruppo FOREIGN KEY (idGruppo) REFERENCES gruppo(idGruppo)

);

CREATE TABLE accetta (

idutente VARCHAR(10),

idGruppo VARCHAR(10),

idCreatore VARCHAR(10),

accettata BOOLEAN,

orario\_richiesta TIMESTAMP,

data\_richiesta DATE,

CONSTRAINT fk\_richiesta\_utente FOREIGN KEY (idutente) REFERENCES richiesta(idRichiedente),

CONSTRAINT fk\_richiesta\_gruppo FOREIGN KEY (idGruppo) REFERENCES gruppo(idGruppo)

CONSTRAINT fk\_richiesta\_creatore FOREIGN KEY (idcreatore) REFERENCES gruppo(idutente)

);