

Utenti (Email(pk), nome)

Utenti\_Opere\_rel(Email(pk),ID\_opera(pk))

Opere(ID\_opera(pk), titolo, descrizione, data)

Opere\_Immagini\_rel(ID\_opera(pk), ID\_immagine(pk))

Opera\_Autore\_rel( ID\_opera(pk), ID\_autore(pk))

Categorie (ID (pk), id\_opera(fk))

Immagini(ID(pk), img, descrizione)

Autori(ID(pk), nome, cognome, data\_nascita, data\_morte, vita)

L’entità “Utenti”:

Il campo nome accetta un massimo di 50 caratteri.

Il campo email accetta soltanto indirizzi email validi.

L’entita “Utenti\_Opere\_rel”:

Il campo Email permette di identificare la relazione tra utente e opera richiesta

Il campo ID\_opera permette di identificare la relazione tra opera e utente che la richiede

L’entità “Opere”:

Il campo ID\_opera è un codice univoco che permette di identificare una singola opera

Il campo descrizione ha una lunghezza variabile.

Il campo data accetta una data nel formato gg-mm-yy.

Il campo titolo accetta una stringa di caratteri di massimo 100 caratteri.

L’entità “Opere\_Immagini\_rel”:

Il campo ID\_opera è un codice univoco che identifica la relazione tra un’immagine e un’opera

Il campo ID\_immagine è un codice univoco che identifica la relazione tra un’opera e un’immagine

L’entità “Opera\_Autore\_rel”:

Il campo ID\_opera è un codice univoco che identifica la relazione tra un’opera ed un autore

Il campo ID\_autore è un codice univoco che identifica la relazione tra un autore ed un’opera

L’entità “Categorie”:

Il campo ID è un codice univoco che rappresenta l’intero nome della categoria a cui appartiene l’opera.

Il campo id\_opera permette il legame 1 a molti tra categorie e opere

L’entità “Immagini”:

Il campo ID è un codice univoco che identifica un’immagine

Il campo img è dedicato al percorso dell’immagine

Il campo descrizione è dedicato ad una breve descrizione dell’immagine nel caso in cui la risorsa non fosse accessibile

L’entità “Autori”:

Il campo ID è un codice univoco che identifica un autore

Il campo nome accetta un massimo di 50 caratteri.

Il campo cognome accetta un massimo di 50 caratteri.

Il campo data\_nascita accetta una data nel formato gg-mm-yy.

Il campo data\_morte accetta una data nel formato gg-mm-yy.

Il campo vita accetta un massimo di 1000 caratteri

La conversione del seguente database in Django è la seguente

Class Utenti (models.Model):

nome = models.CharField(max\_lenght=50)

email = models.EmailField(unique=true, blank=false)

Class Opere (models.Model):

ID\_opera = models.AutoField(primary\_key=True)

descrizione = model.TextField()

data = model.DataField()

titolo = models.CharField(max\_lenght=50)

utenti = models.ManyToManyField(Utenti)

immagini = models.ManyToManyField(Immagini)

Class Immagini (models.Model):

ID = models.AutoField(primary\_key=True)

img = models.ImageField(upload\_to='immagini/')

descrizione = models.TextField()

Class Categorie(models.Model):

ID = models.AutoField(primary\_key=True)

id\_opera = models.ForeignKey(Opere, on\_delete=models.CASCADE)

Class Autori (models.Model):

ID = models.AutoField(primary\_key=True)

nome = models.CharField(max\_lenght=50)

cognome = models.CharField(max\_lenght=50)

data\_nascita = model.DataField()

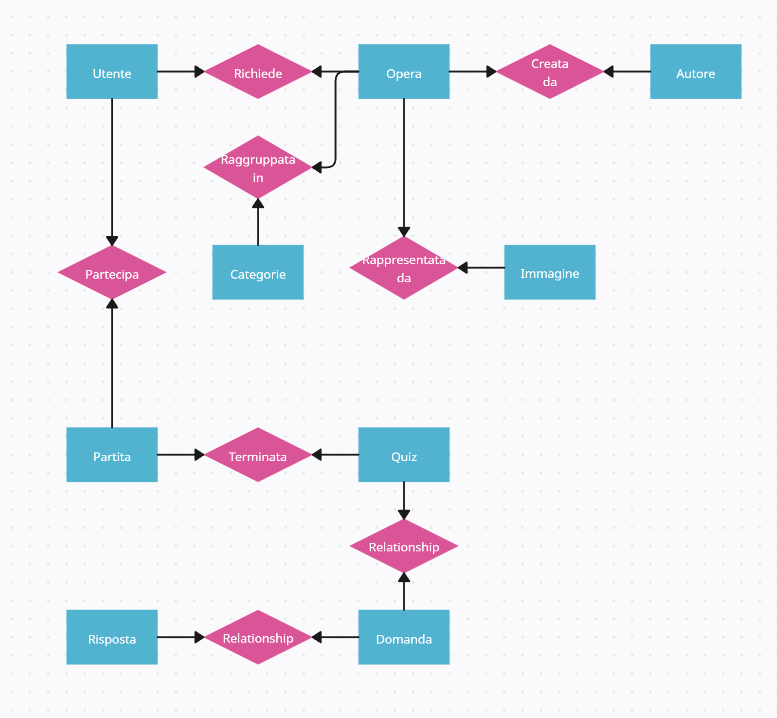
data\_morte = model.DataField()

vita = models.TextField()

opere = models.ManyToManyField(Opere)

—------------------------------------------------------------------------------------------------------------------------

Gamification: Database



Utente(Email(pk), nome)

Partita(ID\_partita(pk), punteggio)

Partita\_Quiz\_Rel(ID\_partita(pk), ID\_quiz(pk), terminato)

Quiz(ID\_quiz(pk), categoria)

Domanda(ID\_domanda(pk), id\_quiz(fk), body, url\_pagina)

Risposta(ID\_risposta(pk), id\_domanda(fk), body, IsRisposta)

L’ Entità “Utente”:

Il campo nome accetta un massimo di 50 caratteri.

Il campo email accetta soltanto indirizzi email validi.

L’ Entità “Partita”:

Il campo ID\_partita è un codice univoco che permette il salvataggio di dati della specifica partita

Il campo punteggio viene calcolato rispetto alla quantità di quiz corretti. Dopo due domande errate la partita finisce e il punteggio finale viene calcolato

L’ Entità “Partita\_Quiz\_Rel”:

Il campo ID\_partita è il codice univoco che lega una certa partita ad un certo quiz

Il campo ID\_quiz è è il codice univoco che lega un certo quiz ad una certa partita

Il campo terminato indica se il quiz è stato finito nella determinata partita

L’ Entità “Quiz”:

Il campo ID\_quiz è un codice univoco che identifica un quiz

Il campo categoria specifica la categoria del quiz

L’ Entità “Domanda”:

Il campo ID\_domanda è un codice univoco che identifica una singola domanda

Il campo id\_quiz permette il legame 1 a molti tra la domanda e i quiz

Il campo body permette la scrittura del testo della domanda

Il campo url\_pagina porta alla pagina contenente i dati necessari per rispondere. Una soluzione spiegata

L’ Entità “Risposta”:

Il campo ID\_risposta è un codice univoco che identifica una singola risposta

Il campo id\_domanda permette il legame 1 a molti tra risposte e domanda

Il campo body permette la scrittura della risposta

Il campo IsRisposta specifica se la risposta scelta è giusta o sbagliata

La conversione del seguente database in Django è la seguente

Class Utenti (models.Model):

nome = models.CharField(max\_lenght=50)

email = models.EmailField(unique=true, blank=false)

Class Partite (models.Model):

ID\_partita = models.AutoField(primary\_key=True)

punteggio = models.IntegerField()

quiz = models.ManyToManyField(Quiz)

Class Quiz (models.Model):

ID\_quiz = models.AutoField(primary\_key=True)

id\_domanda = models.ManyToManyField(Domande)

categoria = models.CharField(max\_lenght=50)

terminato = models.BooleanField(default=False)

Class Domande (models.Model):

ID\_domanda = models.AutoField(primary\_key=True)

id\_risposta = models.ForeignKey(Risposte, on\_delete=models.CASCADE)

body = models.TextField()

url\_pagina = models.URLField()

Class Risposte (models.Model):

ID\_risposta = models.AutoField(primary\_key=True)

body = models.TextField()

IsRisposta = models.BooleanField(default=False)