

# **Esercitazione 2**

Vincoli dichiarativi e store procedure  
in MySQL

# Lo schema logico “esercitazione2”

## (1/3)

### **Corso(nome, n°\_crediti)**

inclusione: Corso[nome]  $\subseteq$  prevede [corso]

### **Studente(matricola)**

Foreign key: Studente[matricola]  $\subseteq$  PianoDiStudio [studente]

### **Sessione\_di\_laurea(data)**

inclusione: SessioneDiLaurea[data]  $\subseteq$  discussione[sessioneDiLaurea]

### **PianoDiStudio(studente, tipo)**

Foreign key: PianoDiStudio[studente]  $\subseteq$  Studente[matricola]

inclusione: PianoDiStudio[studente]  $\subseteq$  prevede [pianoDiStudio]

### **RelazioneFinale(studente, n°pag)**

Foreign key: RelazioneFinale [studente]  $\subseteq$  Studente[matricola]

Foreign key: RelazioneFinale[studente]  $\subseteq$  argomento[relazione]

# Lo schema logico “esercitazione2” (2/3)

## **Esame(studente, corso, voto, data)**

Foreign key: Esame [studente]  $\subseteq$  Studente [matricola]

Foreign key: Esame [corso]  $\subseteq$  Corso [nome]

## **CorsoNonProgettuale(nome)**

Foreign key: CorsoNonProgettuale [nome]  $\subseteq$  Corso [nome]

## **CorsoProgettuale(nome)**

Foreign key: CorsoProgettuale [nome]  $\subseteq$  Corso [nome]

## **Prevede(pianoDiStudio, corso)**

Foreign key: prevede [pianoDiStudio]  $\subseteq$  PianoDiStudio [studente]

Foreign key: prevede [corso]  $\subseteq$  Corso [nome]

# Lo schema logico “esercitazione2” (3/3)

## **Argomento(relazione, corso)**

Foreign key: argomento[relazione]  $\subseteq$  RelazioneFinale[studente]

Foreign key: argomento[corso]  $\subseteq$  Corso[nome]

## **Discussione(relazioneFinale, sessioneDiLaurea, voto)**

Foreign key: Discussione[relazioneFinale]  $\subseteq$  RelazioneFinale[studente]

Foreign key: Discussione[sessioneDiLaurea]  $\subseteq$  SessioneDiLaurea[data]

# Obiettivo 1

**Tradurre lo schema logico su un database MySQL**

Nota: Tralasciare, per ora, tutti i vincoli che non possono essere implementati in maniera dichiarativa nel database MySQL

# Obiettivo 2

## Creare e testare:

1. Una user defined function che, dato un voto di laurea in input (con tipo di dato coerente con la definizione della tabella **Discussione**), restituisca true se il voto di laurea è compreso fra 70 e 110, false altrimenti.
2. Una stored procedure con un parametro di input ed uno di output: il parametro di input rappresenta un piano di studi (con tipo di dato coerente con la definizione della tabella **PianoDiStudio**), il parametro di output è un booleano che rappresenta il risultato della computazione della stored procedure, che è la seguente: se il piano di studio compare almeno una volta nella tabella **Prevede**, allora il valore che viene scritto nel parametro di output è true, altrimenti viene scritto false.
3. Una user defined function, dato l'identificatore di una relazione finale in input (con tipo di dato coerente con la definizione della tabella **RelazioneFinale**), restituisca true se lo studente autore di quella relazione finale può laurearsi, ovvero se per ogni corso previsto dal piano di studi di quello studente esiste un esame, relativo a quel corso e verbalizzato da quello studente, con voto  $\geq 18$ , false altrimenti.