

12/01/2015

Diseño de BBDD

Práctica evaluable: Preguntas para exámenes de tipo test

Emilio Cobos Álvarez (070912324-N)

Diseño de BBDD

Práctica evaluable: Preguntas para exámenes de tipo test Sobre las convenciones utilizadas

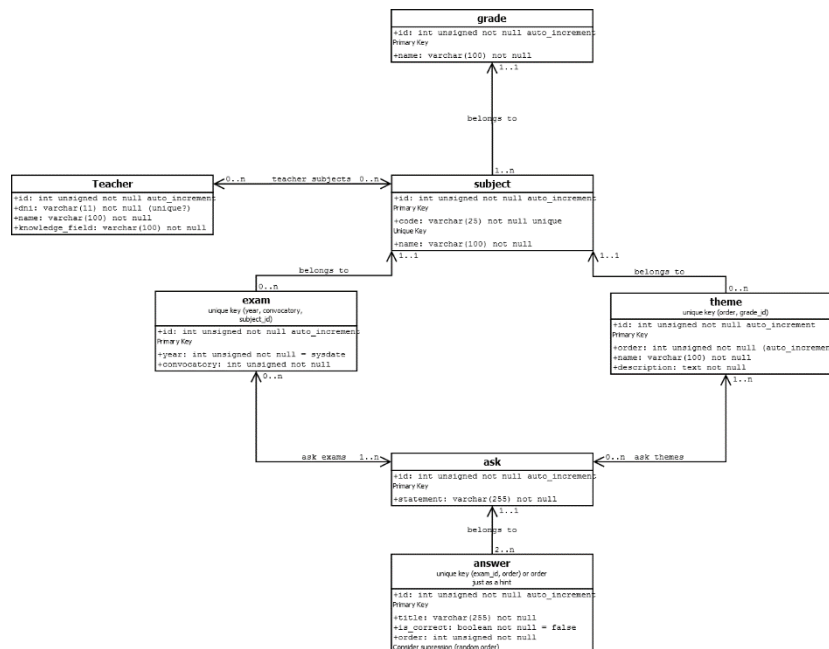
Las convenciones que he utilizado son las siguientes:

- Los atributos se especificarán preferiblemente con minúsculas.
- Si un atributo tiene varias palabras en su nombre, se separarán con barras bajas.
- Las tablas que representan una entidad se llamarán como la entidad, pluralizado.
- Las tablas para relaciones n-m se llamarán con el nombre de la primera entidad ordenada alfabéticamente en singular, seguido de una barra baja, seguido de la otra entidad en plural. Así, la relación entre la entidad *teacher* y la entidad *subject* vendrá definida por la tabla *subject_teachers*.

Diagrama entidad relación

El diagrama entidad relación fue generado con la herramienta GNU Dia. Esta herramienta no deja usar flechas dobles en las asociaciones entre clases UML. Por lo tanto se usará una flecha para asociaciones con cardinalidad máxima n, y no se usarán flecha en el lado de la asociación con cardinalidad máxima 1.

Debajo se encuentra la figura del diagrama en formato PNG. Se adjunta en el fichero de entrega el diagrama original y una versión en SVG.



Paso a relacional

El modelo relacional quedaría así, siguiendo las convenciones anteriormente marcadas:

Las claves primarias se marcarán con un subrayado normal, las claves únicas con un subrayado punteado, y las claves foráneas con un subrayado alterno.

grades:

- id: int unsigned not null auto_increment
- name: varchar(100) not null

subjects:

- id: int unsigned not null auto_increment
- code: varchar(100) not null
- name: varchar(100) not null
- grade_id: int unsigned not null -> *grades(id)*

teachers:

- id: int unsigned not null auto_increment
- dni: varchar(20) not null
- name: varchar(100) not null
- knowledge_field: varchar(100) not null

themes:

- id: int unsigned not null auto_increment
- order: int unsigned not null
- name: varchar(100) not null
- description: text not null
- subject_id: int unsigned not null -> *subjects(id)*
Unique Key (order, subject_id)

asks:

- id: int unsigned not null auto_increment
- statement: varchar(255) not null

answers:

- id: int unsigned not null auto_increment
- title: varchar(255) not null
- is_correct: boolean not null default false
- order: int unsigned not null
- ask_id: int unsigned not null -> *asks(id)*
Unique Key (order, ask_id)

exams:

- id: int unsigned not null auto_increment
- year: int unsigned not null
- convocatory: int unsigned not null

- subject_id: int unsigned not null -> *subjects(id)*
Unique Key (year, convocatory, subject_id)

ask_themes:

- ask_id: int unsigned not null -> *asks(id)*
- theme_id: int unsigned not null -> *themes(id)*
Primary Key (ask_id, theme_id)

ask_exams:

- ask_id: int unsigned not null -> *asks(id)*
- exam_id: int unsigned not null -> *exams(id)*
Primary Key (ask_id, exam_id)

subject_teachers:

- teacher_id: int unsigned not null -> *teachers(id)*
- subject_id: int unsigned not null -> *subjects(id)*
Primary Key (teacher_id, subject_id)

Dependencias funcionales

No es complicado de ver que en todas las tablas que representan entidades las dependencias funcionales son (por diseño) de la forma id -> atributo. Siendo atributo cualquier otro atributo de la tabla.

Es fácil de ver que en todos los casos las tablas están en **FNBC**, ya que el campo id es superclave.

Sobre el código SQL

El código SQL adjunto está realizado para el motor MySQL, por lo que es muy probable que la sintaxis no sea 100% compatible con Oracle.