



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

# **IIC2413: Bases de Datos**

## **Entrega 2**

### **Grupo 5**

(Opción 2)

<b>Alumnos</b>	: Carlos García Felipe Garrido Nicolás Gebauer
<b>Profesor</b>	: Andrés Letelier
<b>Fecha de Entrega</b>	: 24 de septiembre de 2015

# ÍNDICE

<b>I. INTRODUCCIÓN .....</b>	<b>2</b>
<b>II. DIAGRAMA.....</b>	<b>2</b>
<b>III. EXPLICACIÓN DE ENTIDADES Y RELACIONES .....</b>	<b>3</b>
<b>IV. TABLAS GENERADAS .....</b>	<b>4</b>
<b>V. DEPENDENCIAS FUNCIONALES .....</b>	<b>6</b>
<b>VI. CONSULTA SQL.....</b>	<b>6</b>
<b>VII. CONCLUSIONES.....</b>	<b>7</b>

## I. INTRODUCCIÓN

La presente entrega de Bases de Datos tiene por objetivo desarrollar el modelo relacional para el sistema RENNAB. Para lo cual, se diseñará y generará un esquema en BCNF que responda a las necesidades del sistema.

Como grupo se decidió empezar esta entrega desde cero, para lo que se creó un nuevo modelo de Entidad/Relación (E/R) completo del dominio.

En este informe primero se encontrará el nuevo diagrama E/R y luego la explicación de las entidades y relaciones, junto a sus llaves y atributos. Posteriormente, se presentará el algoritmo SQL que se utilizó para generar las tablas. Más tarde, se identificarán las dependencias funcionales y, por último, se mostrará cómo se responde una consulta en SQL a partir de las tablas generadas.

## II. DIAGRAMA

Se decidió empezar la entrega desde cero, para lo cual se diseñó el diagrama E/R de la figura 1

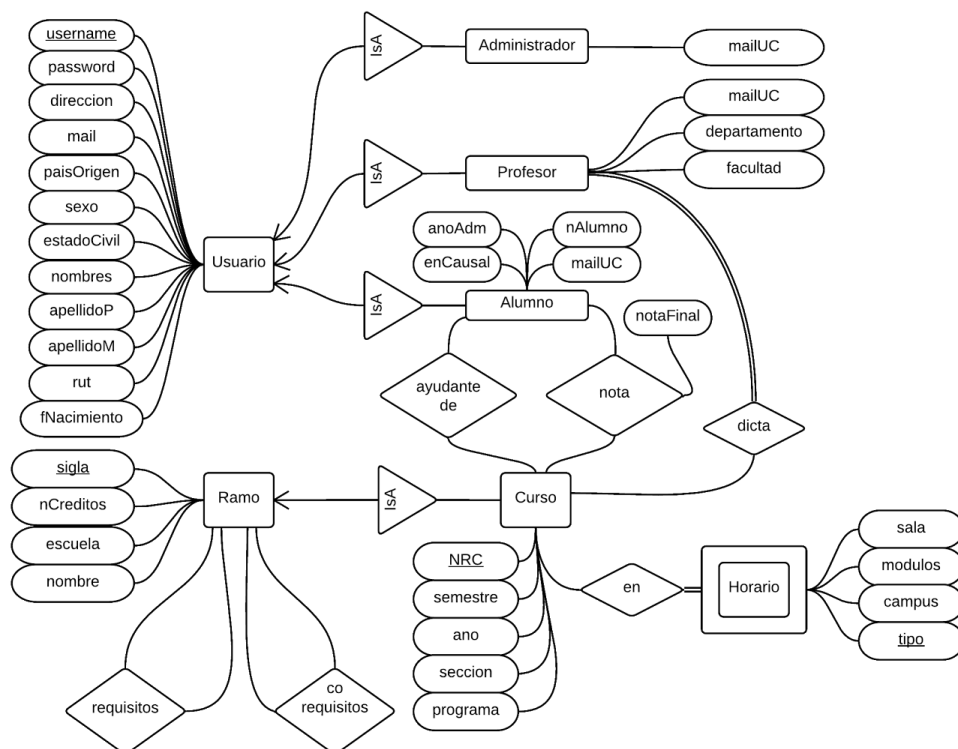


Figura 1: Diagrama E/R

### III. EXPLICACIÓN DE ENTIDADES Y RELACIONES

En el diagrama tenemos las siguientes relaciones:

#### ENTIDADES

- **Usuario:** Cada persona en la universidad y que es parte del sistema RENNAB será un usuario definido por su atributo llave username. Cada usuario tiene, además de su username, una contraseña (password)<sup>1</sup> para autenticarse, nombres (nombres), un apellido paterno (apellidoP), un apellido materno (apellidoM), RUT (rut), Fecha de Nacimiento (fNacimiento), una dirección física (direccion), un mail de contacto (mail), un país de origen (paisOrigen), un género (sexo) y un estado civil (estadoCivil).
- **Alumno<sup>2</sup>:** Cada estudiante en la universidad tendrá un **usuario** asociado a través del atributo llave username. Sus demás atributos son su número de alumno (nAlumno), año de admisión (anoAdmision), mail de la institución (mailUC) y si está o no en causal (enCausal). Siendo un alumno, el usuario tendrá funciones y acceso a otras relaciones, por ejemplo, podrá ser **ayudante de** un curso y tener una **nota**.
- **Profesor<sup>3</sup>:** Cada profesor en la universidad tendrá un **usuario** asociado a través del atributo llave username. Sus demás atributos son mail de la institución (mailUC) y el departamento (departamento) y la facultad (facultad) de la universidad a la que pertenece. Siendo un profesor, el usuario tendrá funciones y acceso a otras relaciones, como por ejemplo **dictar** un **Curso**.
- **Administrador<sup>4</sup>:** Cada administrador del sistema tendrá un **usuario** asociado a través del atributo llave username. Su otro atributo es su mail de la institución (mailUC).
- **Ramo:** Cada ramo de clases en la universidad será identificado por su sigla -la cual tiene 7 caracteres: los primeros 3 corresponden a la facultad y/o departamento de la universidad; los últimos 4 son un número identificador. Además, cada ramo tiene los atributos correspondientes a número de créditos (nCreditos), escuela (escuela), nombre del ramo (nombre). Por último, cada Ramo tiene asociado **Requisitos** (requisitos) y **Corequisitos** (corequisitos).
- **Curso:** Cada curso es parte de un **ramo**, y cada curso es identificado por su NRC. Los atributos de los cursos son su sección (seccion), el año (ano) y el semestre (semestre) en que fue dictado, además la dirección URL de su programa (programa).
- **Horario:** El momento en que se dicta una clase será identificado por la llave del **Curso en** que se dicta -su NRC (nrc)- y también por el tipo de clase que sea -cátedra,

---

<sup>1</sup> El nombre de cada atributo está en paréntesis.

<sup>2,3,4</sup> Alumno, Profesor y Administrador son los roles que pueden llegar a tener los usuarios. En el sistema es posible que puedan tener más de uno, ya que así es más cercano a la realidad.

ayudantía, laboratorio o taller. Sus demás atributos son sala (sala), módulos de clase en que se desarrollan (modulos) y campus (campus).

## RELACIONES

- **Dicta:** Un **Profesor** puede dictar uno o más **Cursos**. Además, cada Curso debe tener al menos un Profesor.
- **En:** Cada **Curso** y **tipo** de clase es dictado en un horario específico
- **Ayudante de:** Un alumno podrá ser ayudante de uno o más **Cursos**.
- **Nota:** Cada **Alumno** al estar tomando un **Curso** tendrá que tener el atributo de una nota final (notaFinal)
- **Requisitos:** Indica qué ramos tienen que ser previamente aprobados para tomar el ramo.
- **Corequisitos:** Indica qué ramos son requisitos y pueden ser tomados simultáneamente al ramo.

## IV. TABLAS GENERADAS

### ENTIDADES

**Usuario** (username varchar(20) PRIMARY KEY<sup>5</sup>, password varchar(20), direccion varchar(100), mail varchar(30), paisOrigen varchar(100), sexo varchar(10), estadoCivil varchar(20), nombres varchar(50), apellidoP varchar(50), apellidoM varchar(50), rut integer check (rut<99999999), fNacimiento date)

**Administrador** (username varchar (20) PRIMARY KEY, mailUC varchar(30), FOREIGN KEY <sup>6</sup>(username) REFERENCES TO Usuario(username) ON DELETE CASCADE)

**Profesor** (username varchar (20) PRIMARY KEY, mailUC varchar(30), departamento varchar(50), facultad varchar(50), FOREIGN KEY (username) REFERENCES TO Usuario(username) ON DELETE CASCADE)

**Alumno** (username varcar (20) PRIMARY KEY, anoAdm integer, nAlumno integer, mailUC varchar(30), enCausal boolean, FOREIGN KEY (username) REFERENCES TO Usuario(username) ON DELETE CASCADE)

---

<sup>5</sup> Las llaves primarias se pueden distinguir con “PRIMARY KEY”

<sup>6</sup> Las llaves foráneas se pueden distinguir con “FOREIGN KEY”

**Curso** (nrc integer PRIMARY KEY, sigla varchar(10), semestre integer, ano integer, seccion integer, programa varchar(100), FOREIGN KEY (sigla) REFERENCES TO Ramo(sigla) ON DELETE CASCADE)

**Ramo** (sigla varchar(10) PRIMARY KEY, nCreditos integer DEFAULT 10, escuela varchar(50), nombre varchar(100))

**Horario** (nrc integer, tipo varchar(15), modulos varchar(20), sala varchar(20), PRIMARY KEY (nrc, tipo), FOREIGN KEY (nrc) REFERENCES TO Curso(nrc) ON DELETE CASCADE)

## RELACIONES

**Nota** (nrc integer, username varchar(20), notaFinal float, PRIMARY KEY(nrc, username), FOREIGN KEY (nrc) REFERENCES TO Curso(nrc) ON DELETE CASCADE, FOREIGN KEY (username) REFERENCES TO Alumno(username) ON DELETE CASCADE)

**ProfesorCurso** (nrc integer, username varchar(20), PRIMARY KEY(nrc, username), FOREIGN KEY (nrc) REFERENCES TO Curso(nrc) ON DELETE CASCADE, FOREIGN KEY (username) REFERENCES TO Profesor(username) ON DELETE CASCADE)

**AyudanteCurso** (nrc integer, username varchar(20), PRIMARY KEY(nrc, username), FOREIGN KEY (nrc) REFERENCES TO Curso(nrc) ON DELETE CASCADE, FOREIGN KEY (username) REFERENCES TO Alumno(username) ON DELETE CASCADE)

**Requisito** (siglacurso varchar(20), siglarequisito varchar(20), PRIMARY KEY(siglacurso, siglarequisito), FOREIGN KEY (siglacurso) REFERENCES TO Ramo(sigla) ON DELETE CASCADE, FOREIGN KEY (siglarequisito) REFERENCES TO Ramo(sigla) ON DELETE CASCADE)

**Corequisito** (siglacurso varchar(20), siglacorequisito varchar(20), PRIMARY KEY(siglacurso, siglacorequisito), FOREIGN KEY (siglacurso) REFERENCES TO Ramo(sigla) ON DELETE CASCADE, FOREIGN KEY (siglacorequisito) REFERENCES TO Ramo(sigla) ON DELETE CASCADE)

## V. DEPENDENCIAS FUNCIONALES

### ENTIDADES

#### Usuario:

username → password, direccion, mail,  
paisOrigen, sexo, estadoCivil, nombres,  
apellidoP, apellidoM, rut, fNacimiento

#### Administrador:

username → mailUC

#### Profesor:

username → mailUC, departamento,  
facultad

#### Alumno:

username → mailUC, anoAdm,  
nAlumno, enCausal

#### Curso:

nrc → sigla, semestre, ano, seccion,  
programa

#### Ramo:

sigla → nCreditos, escuela, nombre

#### Horario:

nrc, tipo → modulos, sala, campus

Como se puede apreciar, cada dependencia funcional está asociada a una llave minimal primaria. Por lo tanto, está en BCNF y no necesitamos hacer una normalización de las tablas.

### RELACIONES

#### Nota:

nrc, username → notaFinal

#### ProfesorCurso:

nrc, username

#### AyudanteCurso:

nrc, username

#### Requisito:

siglacurso → siglarequisito

#### Corequisito:

siglacurso → siglacorequisito

## VI. CONSULTA SQL

**Dado un alumno, muestre todos los cursos que ha aprobado.**

```
SELECT    sigla
FROM      nota, curso
WHERE     notaFinal >= 4.0 AND
          nota.username = 'negebauer' AND
          nota.nrc = curso.nrc;
```

Siendo negebauer el username del alumno del que queremos obtener la lista de cursos aprobados.

## **VII. CONCLUSIONES**

Esta entrega permitió perfeccionar el modelo Entidad/Relación y reconocer la estructura más adecuada para ser implementada en SQL.

La identificación de las llaves primarias y foráneas, junto a las dependencias funcionales indica que el esquema se encuentra en BCNF. Esto es útil al saber que en la base de datos se podrán evitar redundancias y anomalías de inserción y de eliminación en el futuro.

La generación de las tablas y la población de estas en SQL permitieron testear el modelo, y se pudo concluir que está apto para consultas. Esto último pudo ser probado en la sección VI. Consulta SQL

Para terminar esta entrega, se debe esclarecer que nuestro modelo está sujeto a modificaciones a partir de las enseñanzas de cátedra y que colaboren a la integridad de nuestro esquema.