

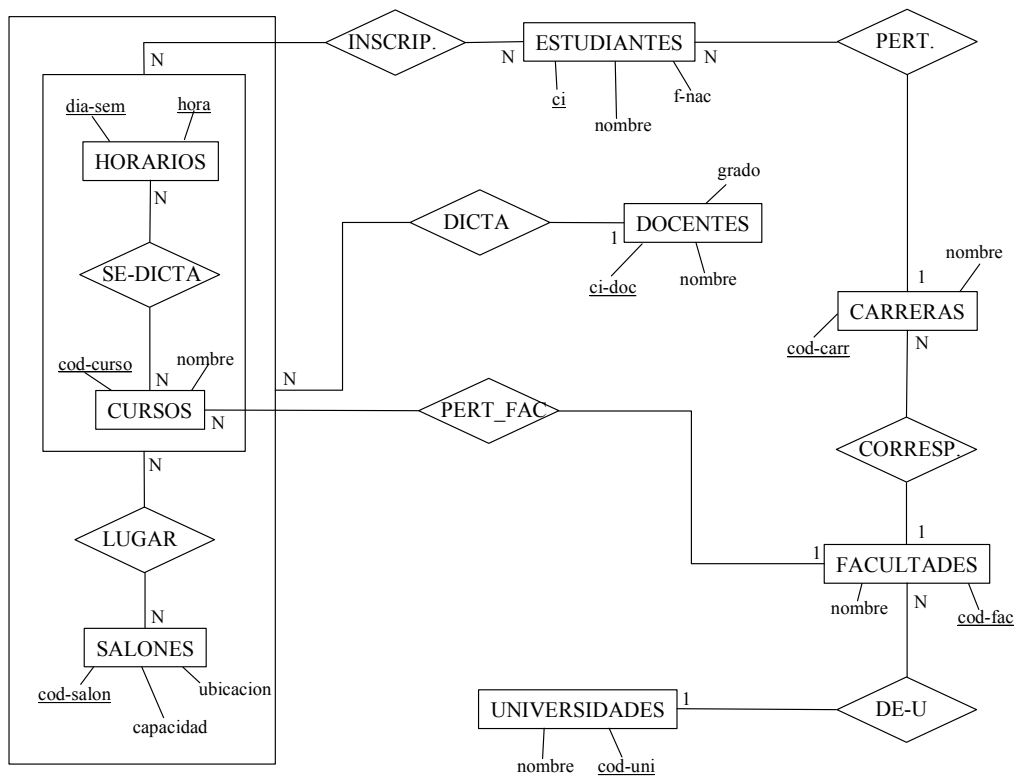
# Práctico de Diseño Lógico Relacional

## Solucion

### Ejercicio 2:

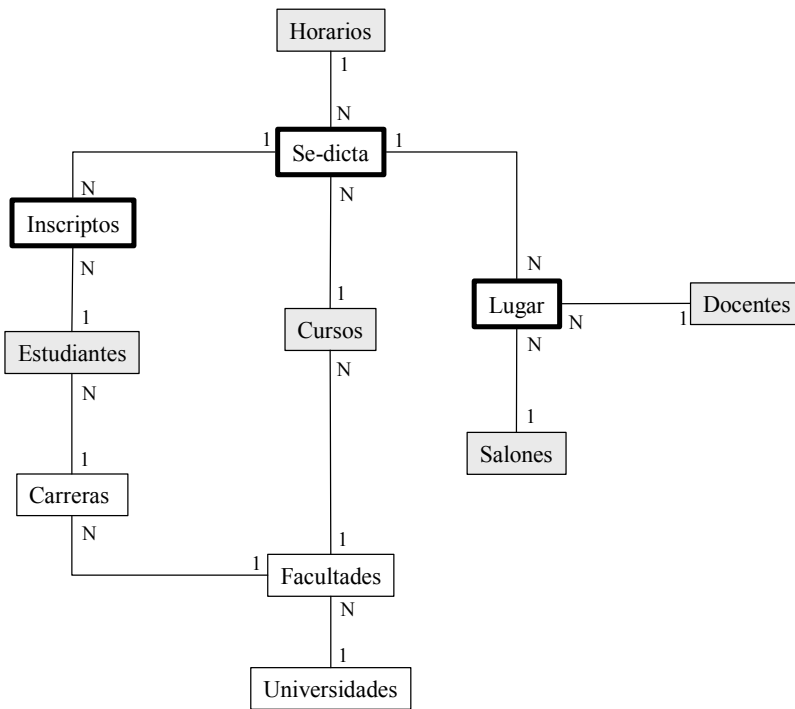
En una facultad se tiene información acerca de cursos que se dictan en salones en distintos horarios. Los cursos pueden corresponder a otras facultades y los estudiantes que se inscriben en los cursos también.

La siguiente es una versión simplificada del esquema Entidad-Relación de la fuente:



Se desea contar con un DW que pueda satisfacer consultas acerca del dictado de cursos, en donde interesan: los cursos, los docentes, los salones, y los estudiantes. Interesa clasificar a los estudiantes por carreras, facultades y universidades, y a los cursos por facultades y universidades.

**Parte a)** Diseñar el esquema lógico relacional del DW utilizando la metodología de [Kortink & Moody, 2000].



### 1) Clasificar entidades

Transaction: Inscriptos, Se-dicta, Lugar

Component: Horarios, Cursos, Salones, Estudiantes, Docentes

Classification: Carreras, Facultades, Universidades

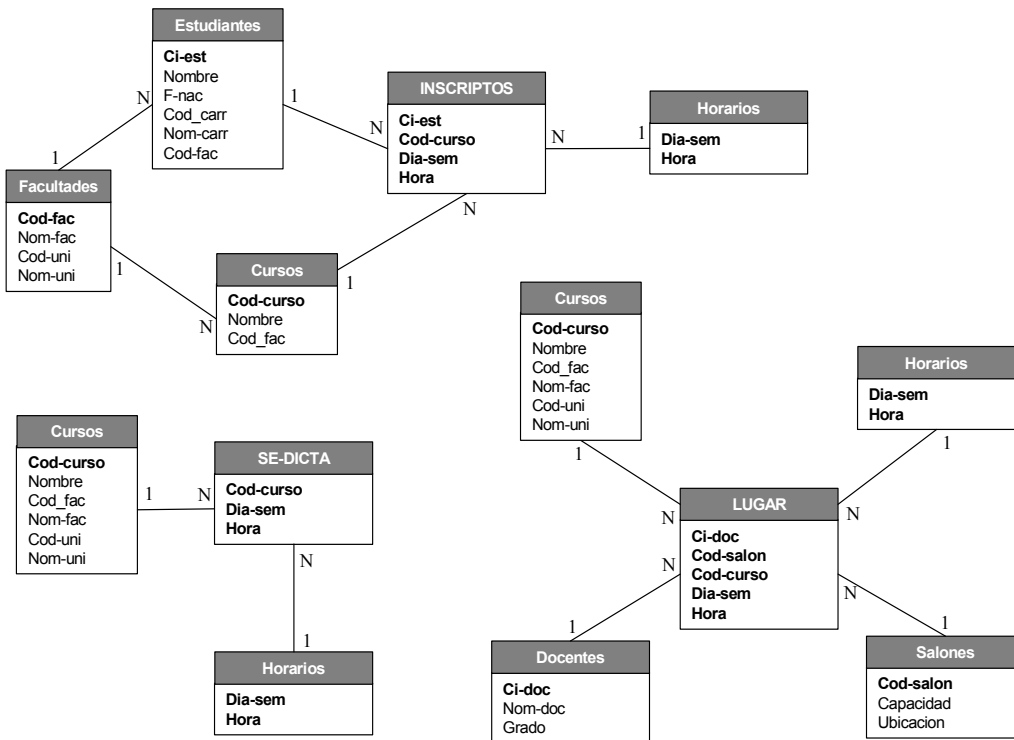
### 2) Identificar jerarquías

- Universidades/Facultades/Carreras/Estudiantes/Inscriptos
- Universidades/Facultades/Cursos/Se-dicta
- Universidades/Facultades/Cursos/Se-dicta/Lugar
- Universidades/Facultades/Cursos/Se-dicta/Inscriptos

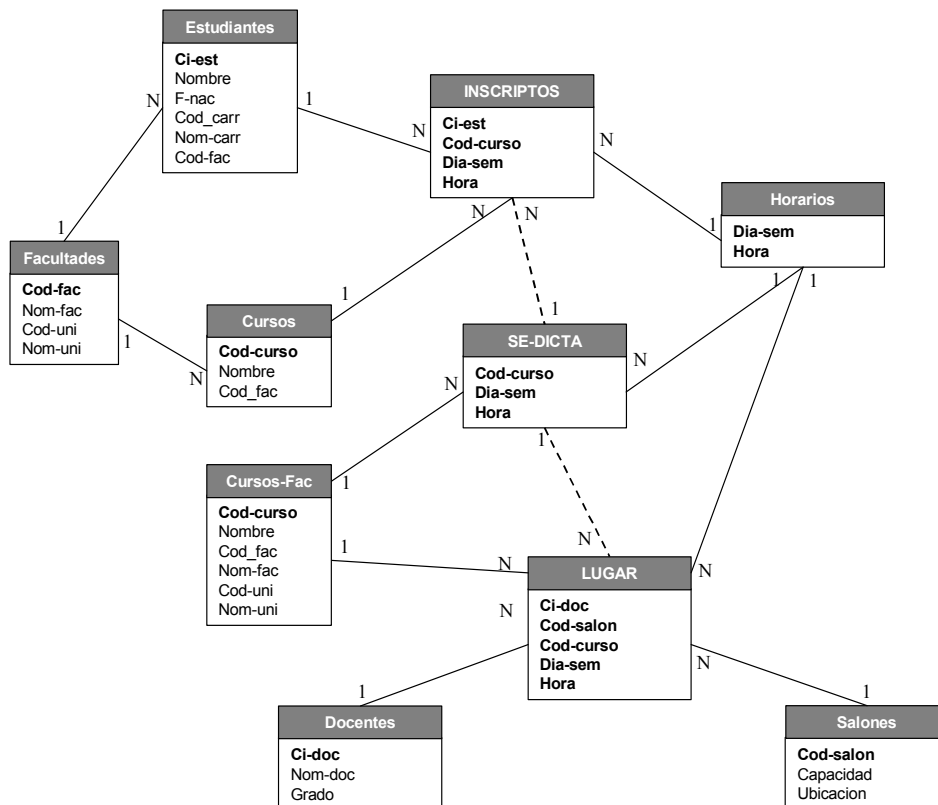
### 3) Colapsar jerarquías

*fork-entity*: Facultades

### Star schema y star-cluster schema:



### Unimos los esquemas:



## Parte b)

- i) Realizar el diseño lógico del DW mediante la aplicación de las primitivas de transformación de esquema. Considerar como esquema objetivo el obtenido en la parte a.

### Esquema Relacional de la base de datos fuente:

Cursos ( cod\_curso, nom\_cur, cod\_fac )  
 Horarios ( dia sem, hora )  
 Se-dicta ( cod\_curso, dia sem, hora )  
 Salones ( cod\_salon, capacidad, ubicacion )  
 Lugar ( cod\_curso, dia sem, hora, cod\_salon, ci\_doc )  
 Docentes ( ci\_doc, nom\_doc, grado )  
 Facultades ( cod\_fac, nom\_fac, cod\_uni )  
 Universidades ( cod\_uni, nom\_uni )  
 Estudiantes ( ci\_est, nom\_est, f\_nac, cod\_carr )  
 Inscriptos ( cod\_curso, dia sem, hora, ci\_est )  
 Carreras ( cod\_carr, nom\_carr, cod\_fac )

**Primitivas a aplicar:**

*IDENTITY* a Inscriptos → Inscriptos-DW

*IDENTITY* a Horarios → Horarios-DW

*IDENTITY* a Cursos → Cursos-DW

*DD-ADDING N-1* a Estudiantes y Carreras → Estud1

*DD-ADDING N-1* a Estud1 y Carreras → Estudiantes-DW

*DD-ADDING N-1* a Facultades y Universidades → Facultades-DW

*IDENTITY* a Se-Dicta → Se-Dicta-DW

*DD-ADDING N-1* a Cursos y Facultades-DW → Cursos01

*DD-ADDING N-1* a Cursos01 y Facultades-DW → Cursos02

*DD-ADDING N-1* a Cursos02 y Facultades-DW → Cursos-Fac-DW

*IDENTITY* a Lugar → Lugar-DW

*IDENTITY* a Salones → Salones-DW

*IDENTITY* a Docentes → Docentes-DW

**APLICACIONES:**

*IDENTITY* a Inscriptos → Inscriptos-DW

R = Inscriptos

**Inscriptos-DW (cod\_curso, dia\_sem, hora, ci\_est)**

.....

*DD-ADDING N-1* a Estudiantes y Carreras → Estud1

R1 = Estudiantes

R2 = Carreras

f = Carreras.nom\_carr

A = cod\_carr

Is\_fk = F

**Estud1 (ci\_est, nom\_est, f\_nac, cod\_carr, nom\_carr)**

*DD-ADDING N-1* a Estud1 y Carreras → Estudiantes-DW

R1 = Estud1

R2 = Carreras

f = Carreras.cod\_fac

A = cod\_carr

Is\_fk = F

**Estudiantes-DW (ci\_est, nom\_est, f\_nac, cod\_carr, nom\_carr, cod\_fac)**

.....

*DD-ADDING N-1* a Cursos y Facultades-DW → Cursos01

R1 = Cursos

R2 = Facultades-DW

f = Facultades-DW.nom\_fac

A = cod\_fac

Is\_fk = F

**Cursos01 (cod\_curso, nom\_cur, cod\_fac, nom\_fac)**

*DD-ADDING N-1* a Cursos01 y Facultades-DW → Cursos02

R1 = Cursos01

R2 = Facultades-DW

f = Facultades-DW.cod\_uni

A = cod\_fac

Is\_fk = F

**Cursos02 (cod\_curso, nom\_cur, cod\_fac, nom\_fac, cod\_uni)**

*DD-ADDING N-1* a Cursos02 y Facultades-DW → Cursos-Fac-DW

R1 = Cursos02

R2 = Facultades-DW

f = Facultades-DW.nom\_uni

A = cod\_fac

Is\_fk = F

**Cursos-Fac-DW (cod\_curso, nom\_cur, cod\_fac, nom\_fac, cod\_uni, nom\_uni)**

.....

ii) Considerar además que:

- a) Las consultas sobre inscriptos son siempre a nivel de carrera y no de estudiante.

**Primitivas a aplicar:**

*HIERARCHY ROLL-UP* a Inscriptos-DW y Estudiantes-DW

**APLICACIONES:**

*HIERARCHY ROLL-UP* a Inscriptos-DW y Estudiantes-DW →

Inscriptos-DW-1 y Estudiantes-DW-1

R1 = Inscriptos-DW, A = ci\_est

R2 = Estudiantes-DW, A = ci\_est

Z = { }

B = cod\_carr

{e1,...,ek} = { }

X = { }

Y = { nom\_est, f\_nac }

agg\_h = T

**Inscriptos-DW-1 (cod\_carr, cod\_curso, dia\_sem, hora)**

**Estudiantes-DW-1 (cod\_carr, nom\_carr, cod\_fac)**

- b) Los docentes van cambiando de grado con el correr del tiempo. Interesa mantener la información acerca de los distintos grados que ha tenido el docente y las fechas en que ha cambiado.

**Primitivas a aplicar:**

*HORIZONTAL PARTITION* a Docentes-DW → Docentes-DW-1 y Docentes-His

*TEMPORALIZATION* a Docentes-His (fecha como parte de la clave)

**APLICACIONES:**

*HORIZONTAL PARTITION* a Docentes-DW → Docentes-DW-1 y Docentes-His

R = Docentes-DW, X = {ci\_doc}

**Docentes-DW-1 (ci\_doc, nom\_doc, grado)**

**Docentes-His (ci\_doc, nom\_doc, grado)**

*TEMPORALIZATION* a Docentes-His → Docentes-His-1

R = Docentes-His

T = fecha

Key = T

**Docentes-His-1 (ci\_doc, fecha, nom\_doc, grado)**