Ejercicios 2da parte

**Normalización**

1) Tenemos los siguientes requerimientos para una base de datos universitaria con que se manejan las boletas de notas de los estudiantes.

Para cada alumno consta su nombre, número de legajo, dirección, teléfono, otro teléfono opcional, fecha de nacimiento, sexo, departamento de carrera, departamento de especialidad.

Para cada departamento, figura su respectivo código con su nombre.

Para cada curso, el nombre del curso, su código, número de horas semanales, nivel, profesor a cargo, año.

Las boletas se conforman del número de legajo, curso, nota y fecha.

A) Diseñar un conjunto de relaciones normalizadas.

B) Llevar todas las relaciones hasta 2FN.

C) Especificar los atributos clave de cada relación.

departamentos (codigo, nombre)

otro\_telefono (legajo\_alumno\*, otro\_telefono)

alumnos (nombre, legajo, direccion, telefono, fecha\_nacimiento, sexo, codigo\_departamento\_carrera\*, codigo\_departamento\_especialidad\*)

profesores (legajo, nombre)

cursos (codigo, nombre, horas\_semanales, nivel, legajo\_profesor\*, anio)

boletas (legajo\*, codigo\_curso\*, nota, fecha)

2) Se trata de hacer una versión normalizada de la estructura que describe una compra de libros a una compañía XZ.

Un pedido consiste en el nombre del cliente, la fecha de pedido, el ISBN (código internacional único) del libro pedido, el título, el autor, la cantidad que ha sido pedida,

y el importe total del período para un libro determinado.

autores (id, nombre)

libros (isbn, titulo, id\_autor\*)

clientes (id, nombre)

pedidos (id, id\_cliente\*, fecha\_pedido, importe\_total)

detalles\_pedidos (id\_pedido\*, isbn\*, cantidad)

3) Tenemos el siguiente conjunto de datos que se va a grabar en una BD de personal de una compañía:

• La compañía tiene un conjunto de departamentos.

• Cada departamento tiene un conjunto de empleados, un conjunto de proyectos y un conjunto de oficinas.

• Cada empleado tiene una historia de empleos. Para cada empleo, tiene una historia de salarios.

• Cada oficina tiene un conjunto de teléfonos.

La base de datos debe contener la siguiente información:

• Para cada departamento: Nro. de departamento (único), presupuesto de cada departamento y número de empleado del gerente del departamento (único).

• Para cada empleado: Nro. de empleado (único), nro. de proyecto actual, nro. de oficina y nro. de teléfono; además,

el título de cada trabajo que ha tenido el empleado, junto con la fecha y el salario para cada salario distinto recibido en ese trabajo.

• Para cada proyecto: Nro. de proyecto (único) y presupuesto del proyecto.

• Para cada oficina: Nro. de oficina (único), área en m2. y números de todos los teléfonos de esa oficina.

Diseñar un conjunto apropiado de relaciones normalizadas hasta 3FN.

empleados (numero, numero\_proyecto\_actual\*, numero\_oficina\*)

departamentos (numero, presupuesto, numero\_empleado\_gerente\*)

proyectos (numero, presupuesto, nro\_departamento\*)

oficinas (numero, area\_en\_m2, nro\_departamento\*)

telefonos (numero, numero\_oficina\*)

historia\_empleos (numero\_hist\_empleo, numero\_empleado\*, titulo, fecha\_desde, fecha\_hasta)

historia\_salarios (numero\_hist\_salario, numero\_hist\_empleo\*, salario)

4) Supongamos que tenemos una relación para proveer la asignación de empleados temporarios a los proyectos:

PROY-ASIGN (NRO-EMP, TEL, SUELDO-HORA, NRO-PROY, FECHA-FINAL)

A) ¿Está en 1FN?

No está en 1FN porque los atributos telefono y sueldo\_hora podrían estar llevados a una tabla 'empleados', ya que podrían no relacionarse directamente con el proyecto asignado.

B) ¿Está en 2FN?

No está en 2FN porque no está en 1FN. Si además existiera el caso de que un empleado puede tener a cargo varios proyectos, la clave primaria debería estar compuesta de numero\_empleado y numero\_proyecto.

C) ¿Está en 3FN?

No está en 3FN porque no está en 2FN.

En caso de que se quiera asignar un empleado a varios proyectos, se debería crear una tabla 'empleados\_proyectos' que contenga la relación entre empleados y proyectos.

O bien asignar como clave primaria de la tabla 'proyecto\_asignado' la combinación de numero\_empleado y numero\_proyecto,

asignando un numero distinto de teléfono, sueldo por hora y fecha final para cada proyecto asignado.

**DER**

Crear el diagrama Entidad / Relación y transformarlo a tablas del modelo relacional.

ENTIDAD ATRIBUTO

Proyectos código

nombre

Empleados legajo

nombre

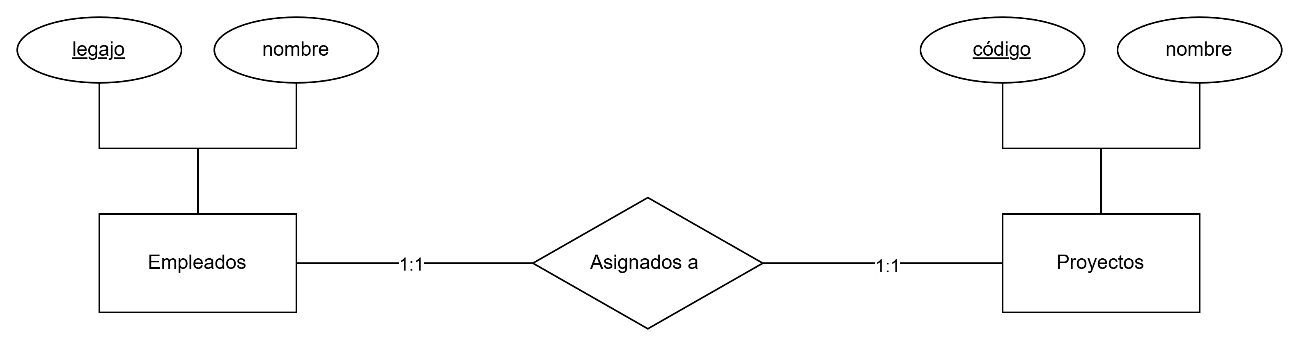
Relación Asignado a

a) Cardinalidad 1 : 1

b) Cardinalidad 1 : N

c) Cardinalidad N : N

A)



proyectos (codigo, nombre, codigo\_empleado\*)

empleados (legajo, nombre, codigo\_proyecto\*)

B)



proyectos (codigo, nombre)

empleados (legajo, nombre, codigo\_proyecto\*)

C)



proyectos (código, nombre)

empleados (legajo, nombre)

proyectos\_empleados (legajo\_empleado\*, codigo\_proyecto\*)

2. Crear el diagrama Entidad / Relación y transformarlo a tablas del modelo relacional.

Entidades:

a) Municipalidades

b) Viviendas

c) Personas

Relaciones:

a) Habita

b) Empadronado en

c) Propietario de

Supuestos datos:

a) Cada persona sólo puede habitar en una vivienda.

b) Cada persona puede ser propietaria de más de una vivienda.

c) Una persona está empadronada en una sola municipalidad.

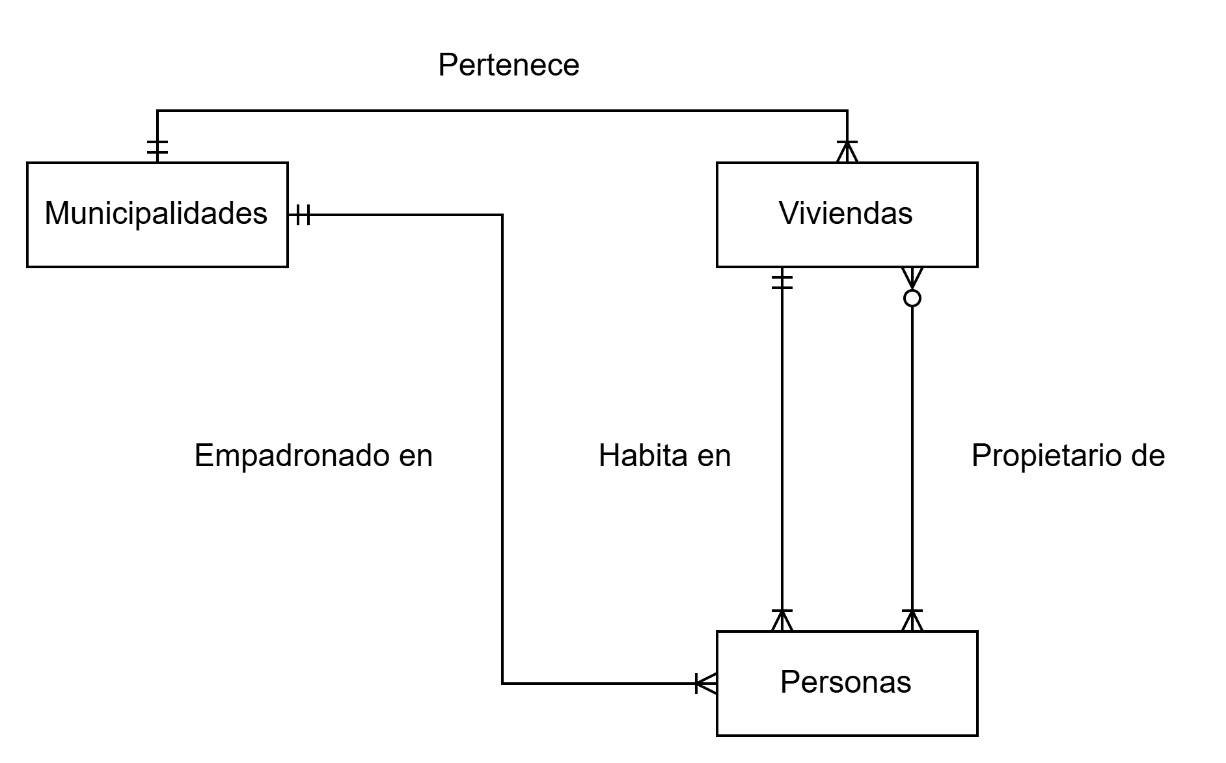
d) En una vivienda pueden habitar más de una persona.

e) Una vivienda puede ser propiedad de más de una persona.

f) Una vivienda pertenece a una sola municipalidad.

g) Una municipalidad puede tener muchas viviendas.

h) En una Municipalidad pueden estar empadronadas muchas personas.



municipalidades (id, nombre)

viviendas (id, direccion, id\_municipalidad\*)

personas (dni, nombre, id\_vivienda\*, id\_municipalidad\*)

propietarios\_viviendas (dni\*, id\_vivienda\*)

3. Crear el diagrama Entidad / Relación y transformarlo a tablas del modelo relacional.

ENTIDAD ATRIBUTO

Pacientes Código\_paciente

Apellido

Dirección

Fecha\_nacimiento

Sexo

Localidad

Empleados Código\_empleado

Apellido

Salario

Localidad

Doctores Código\_doctor

Apellido

Localidad

Salas Código\_sala

Nombre

Numero\_de\_camas

Especialidades Código\_especialidad

Nombre

Funciones Código\_función

Nombre

Turnos Código\_turno

Horario

Relaciones:

Supuestos datos:

a) Un paciente puede estar internado en una sola sala.

b) Un paciente puede ser atendido por varios doctores.

c) En una sala puede haber muchos pacientes internados.

d) Un empleado puede tener asignado una sola función.

e) Un empleado puede trabajar en más de un turno.

f) Una función puede ser cumplida por más de un empleado.

g) Un doctor puede atender a más de un paciente.

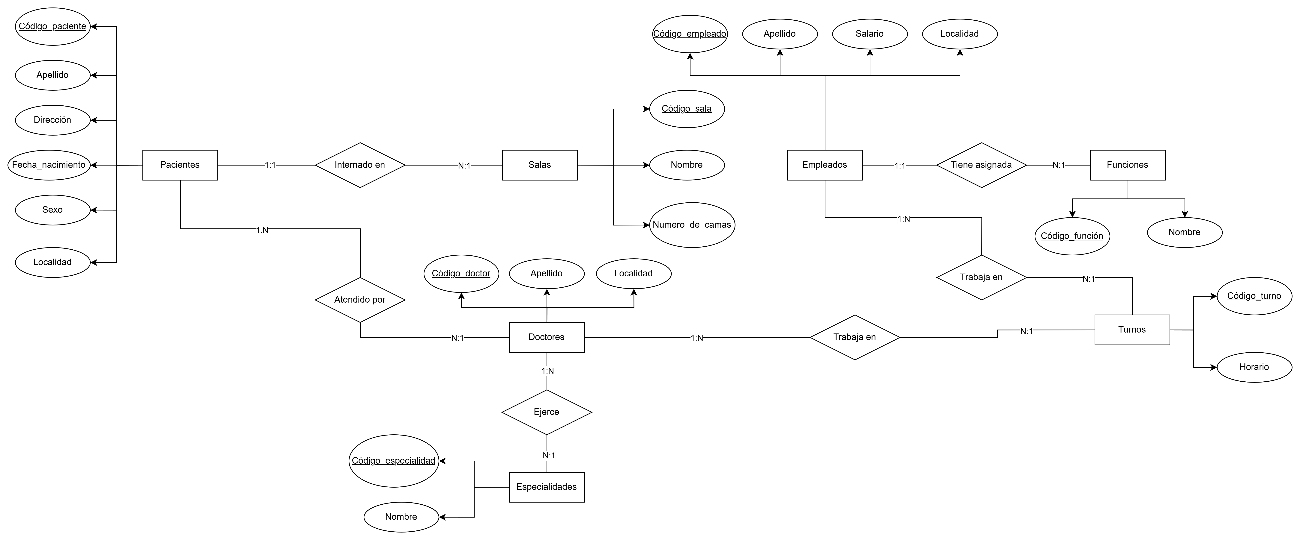
h) Un doctor puede ejercer varias especialidades.

i) Un doctor puede trabajar en más de un turno.

j) Una especialidad puede ser ejercida por más de un doctor.

k) En un turno pueden trabajar más de un empleado.

l) En un turno pueden trabajar más de un doctor.



turnos (codigo, horario)

funciones (codigo, nombre)

especialidades (codigo, nombre)

salas (codigo, nombre, numero\_camas)

doctores (codigo, apellido, localidad)

empleados (codigo, apellido, salario, localidad, codigo\_funcion\*)

pacientes (codigo, apellido, direccion, fecha\_nacimiento, sexo, localidad, codigo\_sala\*)

doctores\_pacientes (codigo\_doctor\*, codigo\_paciente\*)

doctores\_especialidades (codigo\_doctor\*, codigo\_especialidad\*)

doctores\_turnos (codigo\_doctor\*, codigo\_turno\*)

empleados\_turnos (codigo\_empleado\*, codigo\_turno\*)

4. Crear el diagrama Entidad / Relación y transformarlo a tablas del modelo relacional.

El departamento de formación de una empresa desea construir una base de datos para planificar y gestionar la formación de sus empleados.

La empresa organiza cursos internos de formación de los que desea conocer el código de curso, el nombre, una breve descripción, el número de horas de duración y el costo del curso.

Un curso puede tener como prerrequisito haber realizado otro(s) previamente.

Un curso que es un prerrequisito puede serlo de forma obligatoria o sólo recomendable.

Un mismo curso puede ser impartido en diferentes lugares, fechas y con diferentes horarios (día entero, mañana, tarde).

En una misma fecha sólo puede impartirse una edición de un curso.

Los cursos se imparten por personal de la empresa.

De los empleados se desea almacenar su código de empleado, nombre y apellidos, dirección, teléfono, cuit/cuil, fecha de nacimiento, nacionalidad, sexo y salario, así como si está capacitado para impartir cursos.

5. Crear el diagrama Entidad / Relación y transformarlo a tablas del modelo relacional.

La municipalidad de Bariloche desea guardar información sobre las estancias que existen en su jurisdicción y brindan alojamiento a pasajeros. Para ello decide crear una base de datos que contemple las siguientes consideraciones:

Un alojamiento rural (estancia) se identifica por su nombre (“La Tranquila”, “La Rosita”, etc.) que no se repite, tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del establecimiento.

En cada establecimiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el CUIL.

Aunque en un establecimiento trabajen varias personas, una persona puede trabajar en un solo establecimiento.

Los alojamientos se alquilan por habitaciones con una fecha de ingreso y la cantidad de días de permanencia; se desea conocer cuántas habitaciones componen la estancia, de qué tipo (single, doble, triple, etc.) es cada una de ellas, si posee cuarto de baño y el precio diario.

En algunas de estas estancias se realizan actividades organizadas para los huéspedes (senderismo, bicicleta de montaña, etc.). Estas actividades se identifican por un código. Es de interés saber el nombre de la actividad, la descripción y el nivel de dificultad de dicha actividad (de 1 a 10).

Estas actividades se realizan un día a la semana, por ejemplo: en la estancia “La Tranquila” se practica alpinismo los sábados y se desea guardar esta información. Puede haber algún día que no se practique ninguna actividad.

Se pide:

Diseñar el esquema relacional. Indicar las claves primarias y claves externas.

**Seguridad**

1) Crear una vista para un usuario al cual se permite tener acceso a todos los registros de proveedores, pero sin las situaciones. Luego asignarle permisos a un usuario sobre esa vista.

CREATE VIEW V\_PROVEEDORES AS (

SELECT ID\_PROVEEDOR, NOMBRE, DIRECCION, TELEFONO, EMAIL

FROM PROVEEDORES

);

GRANT SELECT ON V\_PROVEEDORES TO USUARIO1;

2) Crear una vista para un usuario al cual se permite tener acceso sólo a los registros de los proveedores situados en Avellaneda, sin las situaciones. Luego asignarle permisos a un usuario sobre esa vista.

CREATE VIEW V\_PROVEEDORES\_AVELLANEDA AS (

SELECT ID\_PROVEEDOR, NOMBRE, DIRECCION, TELEFONO, EMAIL

FROM PROVEEDORES

WHERE LOWER(DIRECCION) = 'avellaneda'

);

GRANT SELECT ON V\_PROVEEDORES\_AVELLANEDA TO usuario2;

3) La tabla DATPERS se define así:

CREATE TABLE DATPERS (

IDENTUSUARIO VARCHAR (8)

SEXO VARCHAR(1),

DEPENDIENTES DECIMAL(2),

OCUPACIÓN VARCHAR(20),

SALARIO DECIMAL(7),

IMPUESTO DECIMAL(7),

AUDITORÍAS DECIMAL(2),

PRIMARY KEY (IDENTUSUARIO)

);

Escribir proposiciones en SQL para conceder lo siguiente:

A) Al usuario Gómez, autorización de selección sobre toda la tabla.

CREATE PUBLIC SYNONYM DATPERS FOR ADMIN.DATPERS;

GRANT SELECT ON DATPERS TO Gómez;

B) Al usuario López, autorización de inserción y eliminación sobre toda la tabla.

GRANT INSERT, DELETE ON DATPERS TO López;

C) Al usuario Sánchez, autorización de selección sobre toda la tabla y autoridad de actualización sobre los campos SALARIO e IMPUESTO (solamente).

GRANT SELECT, UPDATE (SALARIO, IMPUESTO) ON DATPERS TO Sánchez;

D) Al usuario Torres, autorización de selección sobre los campos IDENTUSUARIO, SALARIO e IMPUESTO (únicamente).

CREATE VIEW V\_IDENTUSUARIO\_SALARIO\_IMPUESTO AS (

SELECT IDENTUSUARIO, SALARIO, IMPUESTO

FROM DATPERS

);

CREATE PUBLIC SYNONYM V\_IDENTUSUARIO\_SALARIO\_IMPUESTO FOR ADMIN.V\_IDENTUSUARIO\_SALARIO\_IMPUESTO;

GRANT SELECT ON V\_IDENTUSUARIO\_SALARIO\_IMPUESTO TO Torres;

E) Al usuario Pérez, autorización de selección igual a la de Torres y autorización de actualización sobre los campos Salario e Impuesto (solamente).

GRANT SELECT ON V\_IDENTUSUARIO\_SALARIO\_IMPUESTO TO Pérez;

GRANT UPDATE (SALARIO, IMPUESTO) ON DATPERS TO Pérez;

F) Al usuario Villalba, autorización de selección sobre los registros de predicadores (únicamente). (Predicadores es un tipo de ocupación).

CREATE VIEW V\_PREDICADORES AS (

SELECT IDENTUSUARIO, SEXO, DEPENDIENTES, OCUPACIÓN, SALARIO, IMPUESTO, AUDITORÍAS

FROM DATPERS

WHERE LOWER(OCUPACIÓN) = 'predicadores'

);

CREATE PUBLIC SYNONYM V\_PREDICADORES FOR ADMIN.V\_PREDICADORES;

GRANT SELECT ON V\_PREDICADORES TO Villalba;

G) Al usuario Giménez, autorización de selección igual a la de Torres y autoridad de actualización sobre los campos IMPUESTO y AUDITORIA (únicamente).

GRANT SELECT ON V\_IDENTUSUARIO\_SALARIO\_IMPUESTO TO Giménez;

GRANT UPDATE (IMPUESTO, AUDITORÍAS) ON DATPERS TO Giménez;

H) Al usuario Agüero, autorización de selección sobre los salarios máximos y mínimos por clase de ocupación, pero ninguna otra autorización.

CREATE VIEW V\_SALARIOS\_MAX\_MIN AS (

SELECT OCUPACIÓN, MAX(SALARIO) AS SALARIO\_MAX, MIN(SALARIO) AS SALARIO\_MIN

FROM DATPERS

GROUP BY OCUPACIÓN

);

CREATE PUBLIC SYNONYM V\_SALARIOS\_MAX\_MIN FOR ADMIN.V\_SALARIOS\_MAX\_MIN;

GRANT SELECT ON V\_SALARIOS\_MAX\_MIN TO Agüero;