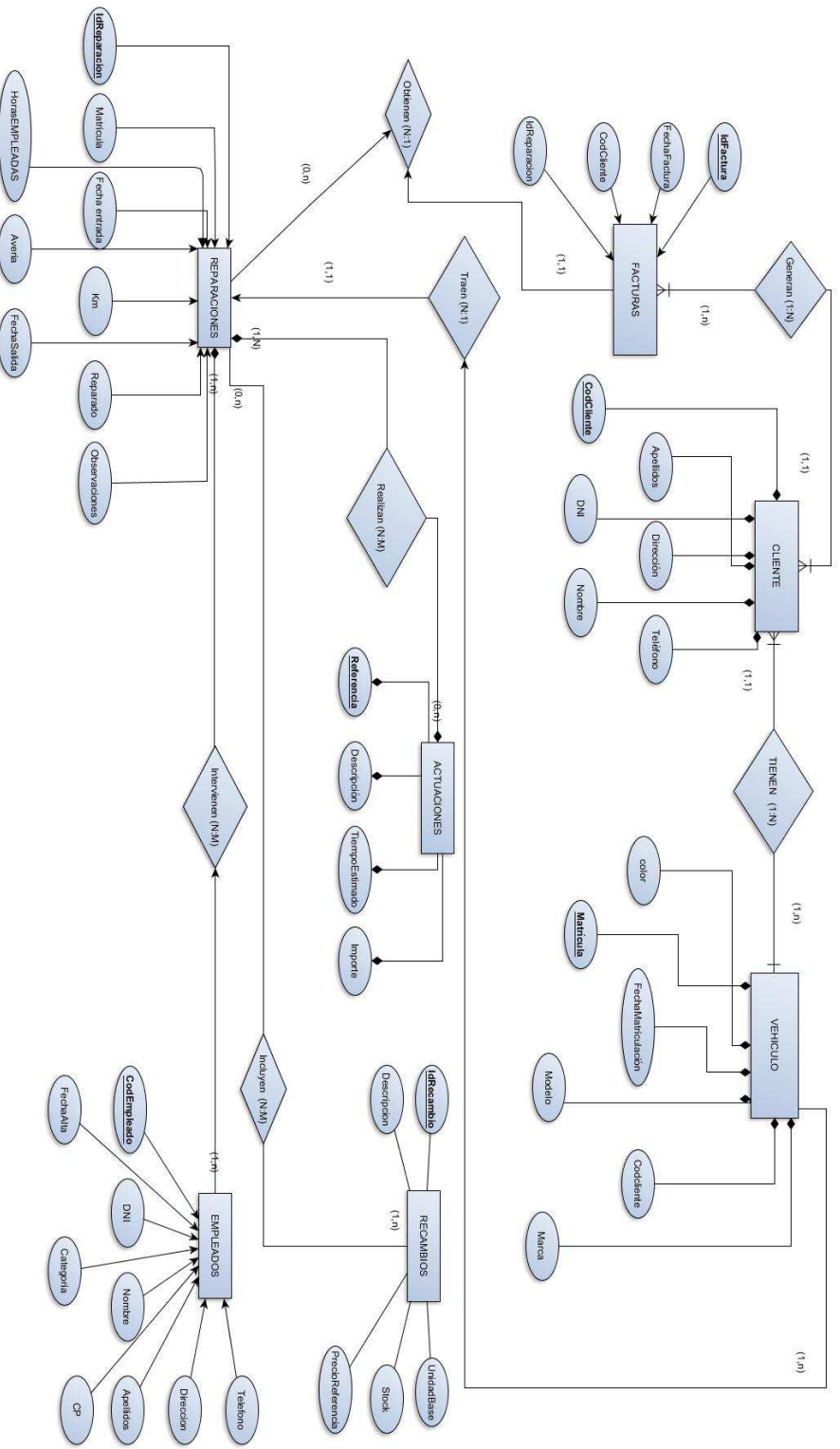


TASK 1:

Entity Relationship-Model:



7 initially define entities (CLIENTE, VEHICULO, FACTURAS, REPARACIONES, EMPLEADOS, RECAMBIO Y ACTUACIONES).

The degree of relations between entities is of second degree because relationships not involving in more than two entities.

The possibility about binary relations correspondence are three: 1: 1, 1: N, N: M.

There are 3 relationships N: M will result tables in the relational model whose

main keys are the keys of the entities that make up the relationship. Furthermore 3

N: M have their own attributes, which in any case also generate a new

table in the relational model but the correspondence was 1: 1 or 1: N.

I put some examples:

Relation between CLIENTE and VEHICULO:

A CLIENTE may have one or various VEHICULOS, and a VEHICULO belongs to a CLIENTE.

Therefore, placing the vehicle view of a VEHICULO can obtain belong at least one client and at most to a CLIENTE, accounting for a cardinality (1,1).

Placing the CLIENTE obtain views on this may have at least one VEHICULO and as Maximum number of vehicles being the cardinality at this end (1, N).

The maximum cardinality, 1 and N defines the degree of the relation in this case is (1, N).

Relation REPARACION and EMPLEADOS:

In a REPARACION they may have involved one or more EMPLEADOS, and an EMPLEADO may be involved in one or several REPARACIONES. "

Therefore, placing the view on REPARACION obtain a repair may have been make for at least one EMPLEADO and a maximum of several EMPLEADOS, establishing cardinality (1, N). Placing the view over EMPLEADOS get this may have been involved in at least one REPARACION and at most several REPARACIONES, establishing new cardinality (1, N).

The maximum cardinality of both ends are N and N, defined in this case the degree of relationship of several to several N: M.

OBTAINING THE RELATIONAL MODEL:

Relation between **CLIENTES** and **FACTURAS** is of one to various 1:N.

The key field in the **CLIENTES** table (**ClientesID**) must be on the table

FACTURAS as foreign key.

The relationship between the **CLIENTE** entities and **VEHICULOS** is one to multiple 1: N, so the key field in the **CLIENTE** table (**CodCliente**) must be on the table **VEHICULOS** as foreign key.

The relationship between the **REPARACIONES** and **FACTURAS** entities is one to one 1: 1. In this if we propagate in the **IdReparación** **FACTURAS** table key, since data for the elaboration of the **FACTURAS** will be obtained from the already obtained in the table **REPARACIONES**.

The relationship between entities **REPARACION** and **VEHICULOS** is one to several 1: N, so the key field of the table **VEHICULOS** (**Matrícula**) must be included in the **REPARACIONES** table as foreign key.

The "Intervienen" relationship defined between the **REPARACION** entities and **EMPLEADOS**, is several to several N: M, so that in the relational model will result in a new table whose keys are the main keys of both entities **CodEmpleado**, **IdReparación**, and also it must include the attribute of the relationship, "**Horas**".

The relationship "**Incluyen**" defined between entities **REPARACIÓN** and **RECAMBIOS** is too many to many N: M, so that in the relational model will lead to a new table whose keys are the main keys of both entities, **IdRecambio**, **IdReparación**, and also it must include the proper attribute "**Unidades**".

The "**Realizan**" relationship defined between the **REPARACION** and **REPARACION** is too many to many N: M, so that in the relational model will lead to a new table whose keys are the main keys of both entities, **IdReparación**, **Referencia**, and also it must include the attribute of the relationship, "**Horas**".

It is important to note that the three relationships "Intervienen", "Incluyen" and "Realizan" would also place tables in the relational model was even one to many as 3 have attributes, which would generate a new table.

With the foregoing, the relational model that remains is:

Relational Model:

CLIENTES (CodCliente, DNI, Apellidos, Nombre, Dirección, Teléfono)
FACTURAS (IdFactura, FechaFactura, CodCliente, IdReparación)
VEHICULOS (Matricula, Marca, Modelo, Color, FechaMatriculación, CodCliente)
REPARACIONES (IdReparación, Matricula, FechaEntrada, Km, Avería, FechaSalida, Reparo, Observaciones, HorasEMPLEADAS)
EMPLEADOS (CodEmpleado, DNI, Nombre, Apellidos, Dirección, Teléfono, CP, FechaAlta, Categoría)
RECAMBIOS (IdRecambio, Descripción, UnidadBase, Stock, PrecioReferencia)
ACTUACIONES (Referencia, Descripción, TiempoEstimado, Importe)
Intervienen (CodEmpleado, IdReparación, Horas)
Incluyen (IdRecambio, IdReparación, Unidades)
Realizan (IdReparación, Referencia, Horas)

TASK 2:

1FN:

We will have to eliminate repetitive groups, if any. We will study the dependencies between the attributes for choosing an appropriate key.

NumPedido + CodProducto

FechaPedido, CodProveedor, NombreProveedor, DirecciónProveedor, NombreProducto, PrecioProducto, Cantidad.

PEDIDO (NumPedido, FechaPedido, CodProveedor, NombreProveedor, DirecciónProveedor, CodProducto, NombreProducto, PrecioProducto, Cantidad)

Already in 1FN.

- It has a defined key.
- No repeating groups.
- All attributes dependent of the primary key.

2FN:

To be in 2NF there can be no functional dependencies of attributes on any part of the primary key.

We have to see if there are partial dependencies. Functional dependencies of non-key attributes in relation to the key are:

NumPedido + CodProducto , Cantidad

NumPedido, FechaPedido, CodProveedor, NombreProveedor, DirecciónProveedor

CodProducto, NombreProducto, PrecioProducto

For it is in 2NF we will have to remove these dependencies breaking tables:

PEDIDO-PRODUCTO (NumPedido, CodProducto, Cantidad)

PEDIDO (NumPedido, FechaPedido, CodProveedor, NombreProveedor, DirecciónProveedor)

PRODUCTO (CodProducto, NombreProducto, PrecioProducto)

Already in 2FN.

- **No partial dependencies. No attribute depends on a part of the primary key.**
- **It is in 1FN.**

3FN:

To this must be eliminated in 3NF transitive dependencies between attributes that are not key.

In the relationship dependencies exist PEDIDO:

NumPedido

FechaPedido

CodProveedor

NombreProveedor, DirecciónProveedor

PEDIDO-PRODUCTO (NumPedido, CodProducto, Cantidad)

PEDIDO (NumPedido, FechaPedido, CodProveedor)

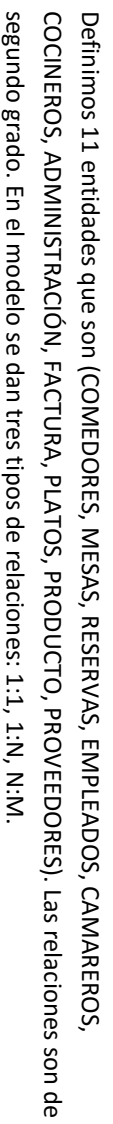
PRODUCTO (CodProducto, NombreProducto, PrecioProducto)

PROVEEDOR (CodProveedor, NombreProveedor, DirecciónProveedor)

Already in 3FN.

- **All is in 2NF**
- **No transitive functional dependencies.**

Entity Relationship-Model:



Relación entre **COMEDORES** and **MESAS** es de 1:N por lo que el campo clave de la tabla **COMEDORES** (**CodComedor**) debe estar incluido en la tabla **MESAS** como clave ajena.

La relación "**Ajudican**" entre las entidades **RESERVAS** y **MESAS** es de varios a varios N:M por lo que el modelo relacional dará lugar a una nueva tabla cuyas claves son las claves principales de ambas entidades **CodComedor**, **NumReserva**, **NumMesa** y que además debe incluir el atributo propio de la relación, "**Plazas**".

Relación entre **EMPLEADOS** y **MESAS** es de 1:N por lo que el campo clave de la tabla **EMPLEADOS** (**CodEmpleado**) debe estar incluido en la tabla **MESAS** como clave ajena.

Para evitar valores nulos en la entidad **EMPLEADOS** se emplea jerarquías de generalización. Teniendo la entidad supertipo **EMPLEADOS** con los atributos comunes (**CodEmpleado**, **Dni**,...**Nombre**, **etc...**) y las entidades subtipo (**CAMAREROS**, **COCINEROS**, **ADMINISTRACIÓN**) que heredan los atributos del supertipo. Las cardinalidades del supertipo **EMPLEADOS** es de (1,1) y las de los subtipos de (0,1).

Relación entre **MESAS** y **FACTURA** es de 1:N por lo que el campo clave de la tabla **MESAS** (**NumMesa** + **CodigoComedor**) debe estar incluido en la tabla **FACTURA**.

La relación "Incluyen" entre las entidades **FACTURA** y **PLATOS** es de varios a varios N:M por lo que el modelo relacional dará lugar a una nueva tabla cuyas claves son las claves principales de ambas entidades (**IdFactura**, **CodPlato**) y que además debe incluir el atributo propio de la relación, "**Unidades**".

La relación "Contienen" entre las entidades **PLATOS** y **PRODUCTO** es de varios a varios N:M por lo que el modelo relacional dará lugar a una nueva tabla cuyas claves son las claves principales de ambas entidades (**CodPlato**, **CodProducto**) y que además debe incluir el atributo propio de la relación, "**Cantidad**".

La relación "Suministran" entre las entidades **PRODUCTO** y **PROVEEDORES** es de varios a varios N:M por lo que el modelo relacional dará lugar a una nueva tabla cuyas claves son las claves principales de ambas entidades (**CodProveedor**, **CodProducto**) y que además debe incluir el atributo propio de la relación, "**Cantidades**".

Relational Model:

EMPLEADOS (CodEmpleado, DNI, AyudTemporal, Categoría, Salario, Nombre, Apellidos, Dirección, Ciudad, CP, Teléfono, FechaAlta)

COCINEROS (Puesto, Especialidad)

ADMINISTRACIÓN (Cargo)

CAMAREROS (Turno, AñosExperiencia)

COMEDORES (CodComedor, Nombre, NumMesa, NumMaxComensales, Localización)

RESERVAS (NumReserva, FechaHora, FechaHoraReserva, NumPersona, NomPersona, DatosInteres, ComidaCena)

Adjudican (NumReserva, NumMesa, CodComedor, Plazas)

MESAS (NumMesa, CodComedor, CodEmpleado, NumAsientos)

FACTURA (IdFactura, NumFactura, Fecha, NumMesa, CodPlato)

Incluyen (IdFactura, CodPlato, Unidades)

PLATOS (CodPlato, NomPlato, TipoPlato, Descripción, Precio, CodProducto)

Contienen (CodPlato, CodProducto, Cantidad)

PRODUCTO (CodProducto, PrecioUnidad, Categoría, Descripción, Stock, UnidadBase)

Suministran (CodProveedor, CodProducto, Cantidades)

PROVEEDORES (CodProveedor, Nombre, Dirección, CP, Telefono, Fax, PersonaContacto, CodProducto)