



Programa Ejecutivo de Business Intelligence

Fundamentos y modelamiento de Base de Datos

Ing. Arturo Rojas Medrano

PROPOSITO GENERAL DE LAS BASES DE DATOS

- Las personas requerimos de las bases de datos para la vida diaria.
- De donde viene la necesidad de almacenar las cosas.
- Las personas hemos tenido necesidad de registrar y almacenar la información, por ejemplo.
 - Papiros
 - Dibujos
 - etc



PROPOSITO GENERAL DE LAS BASES DE DATOS

- ¿Por qué debo almacenar la información ?



Necesito tenerla en el tiempo.

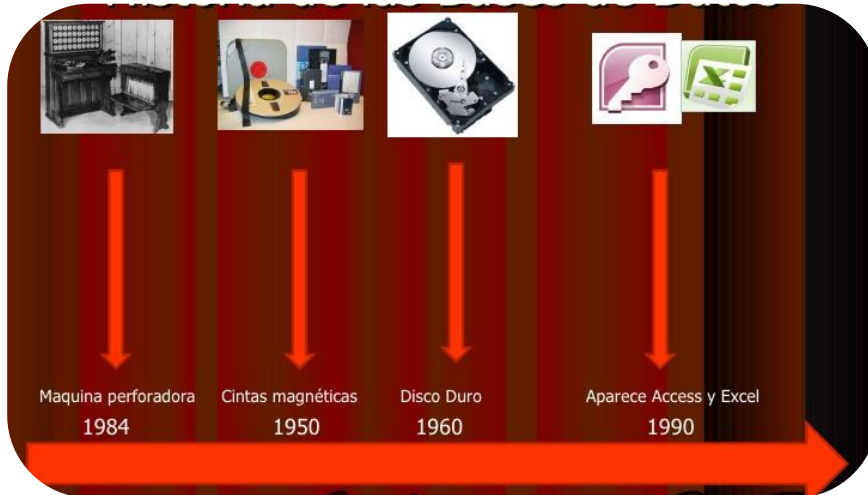
Por ejemplo, los Sistema de archivos en papel, los puedo acceder en el tiempo, pero no son eficientes

PROPOSITO GENERAL DE LAS BASES DE DATOS



- SGBD -> Sistema de Gestión de base de datos.
- Es una estrategia y una plataforma tecnológica, que tiene ciertas operaciones y estrategias para lograr persistir en el tiempo y sacar información de los datos.

HISTORIA DE LAS BASES DE DATOS



- 1950-1960: Maquinas tabuladoras, tarjetas perforadas y cintas magnéticas., eje censos poblacionales
- 1960-1979: Modelos jerárquicos, discos duros, modelo de data relacional, transacciones en tiempo real.



HISTORIA DE LAS BASES DE DATOS

- Un disco duro tiene información persistente, o sea que perdura en el tiempo.
- 1970-1980: SQL, Sistemas SQL comerciales, bases de datos paralelas y distribuidas, bases de datos orientadas a objetos.
- SQL es un estándar, la mayoría de los comandos básicos, en cualquier tipo de datos que sea SQL deben funcionar (MariaDB, MySQL, etc).



HISTORIA DE LAS BASES DE DATOS

- 1980-1990: Data mining, data warehouse, e-commerce.
- 2000-Actualidad: XML, administración automatizada, analytics, big data, No SQL, InMemory, Scale Out, Systems of Engagement.

TIPOS DE BASE DE DATOS

- Bases de Datos Relacionales
- Bases de Datos no Relacionales
- Bases de Datos Relacionales empresariales (más importantes)



NoSQL



TIPOS DE BASE DE DATOS

- Bases de Datos no Relacionales:
 - Bases de datos “Clave” – “Valor”: Es el modelo de NoSQL más popular y sencilla en cuanto a funcionalidad.
 - Bases de datos “Documentales”: Este tipo es el más versátil ya que guarda información como un documento generalmente de tipo JSON o XML.
 - Bases de datos “En grafo”: La información representada en este tipo de bases de datos se realiza en forma de nodos de un grafo y sus relaciones con las aristas del mismo.
 - Bases de datos “Orientadas a Objetos”: En este tipo de bases de datos la información se representa mediante objetos de igual forma que lo hacen los lenguajes de programación orientados a objetos.

APLICACIONES EN LA INDUSTRIA

- ¿ Donde encontramos base de datos ?



VISION DE LOS DATOS

- ¿Qué es un dato?





VISION DE LOS DATOS

- ¿Qué es un dato?
 - Un número
 - Una palabra
 - Un adjetivo
 - Una característica
 - Una imagen
 - Una transacción
 - Una estadística
 - Un código

VISION DE LOS DATOS

- ¿Qué es un dato?
- Un dato es algo que nos va a permitir describir un objeto. Ese objeto global lo vamos a poder llamar “Entidad”. Una entidad puede estar llena de datos.
- Un dato no representa nada, solamente es algo que perdura en el tiempo, sirve para llevar un registro, pero como tal no significa nada, si no lo exploto o no lo consulto de forma correcta, por lo tanto los datos tienen un proceso para ser convertido en información.

VISION DE LOS DATOS

Tres niveles de Abstracción:

- Conceptual: Se tiene que empezar a modelar una Base de Datos dependiendo de lo que se quiere hacer basado en los conceptos de “entidad” y “relación”.
- Lógico: El diagrama lógico nos va a resolver ciertas dudas de consistencia, para evitar crear loops o evitar que tenga cosas que no tengan sentido en nuestro proyecto.
- Físico: Es finalmente cómo lo va a ver la Base de Datos.

Coceptual



Lógico

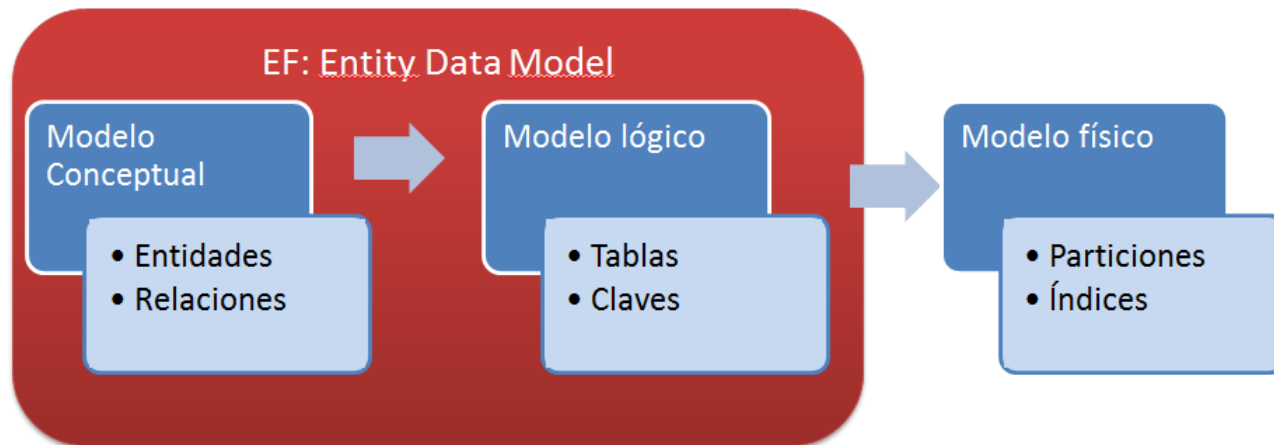


Físico



VISION DE LOS DATOS

Tres niveles de Abstracción:



Coceptual



Lógico



Físico



Igual que en cualquier lenguaje de programación, existen variables en las Bases de Datos:

- Caracteres: Pueden ser desde letras hasta caracteres especiales.
- Numérico: Del 0 al 9 pero con una longitud especial.
- Varchar: Caracteres con un formato más variable.
- Imagen
- Fecha: Generalmente van acompañadas de una hora.
- Moneda: esto facilita todo si se trabaja con diferentes denominaciones.
- Texto: Variables que tienen mayor tamaño que un char o que un varchar.
- Bit: Se puede trabajar con 1 y 0 o también con verdadero y falso.
- Decimal

- **Esquema:**
 - Es la estructura lógica que va a tener una Base de Datos.
 - Define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.
- **Instancia:**
 - Es el conjunto de datos que tiene una Base de Datos en un instante de tiempo.

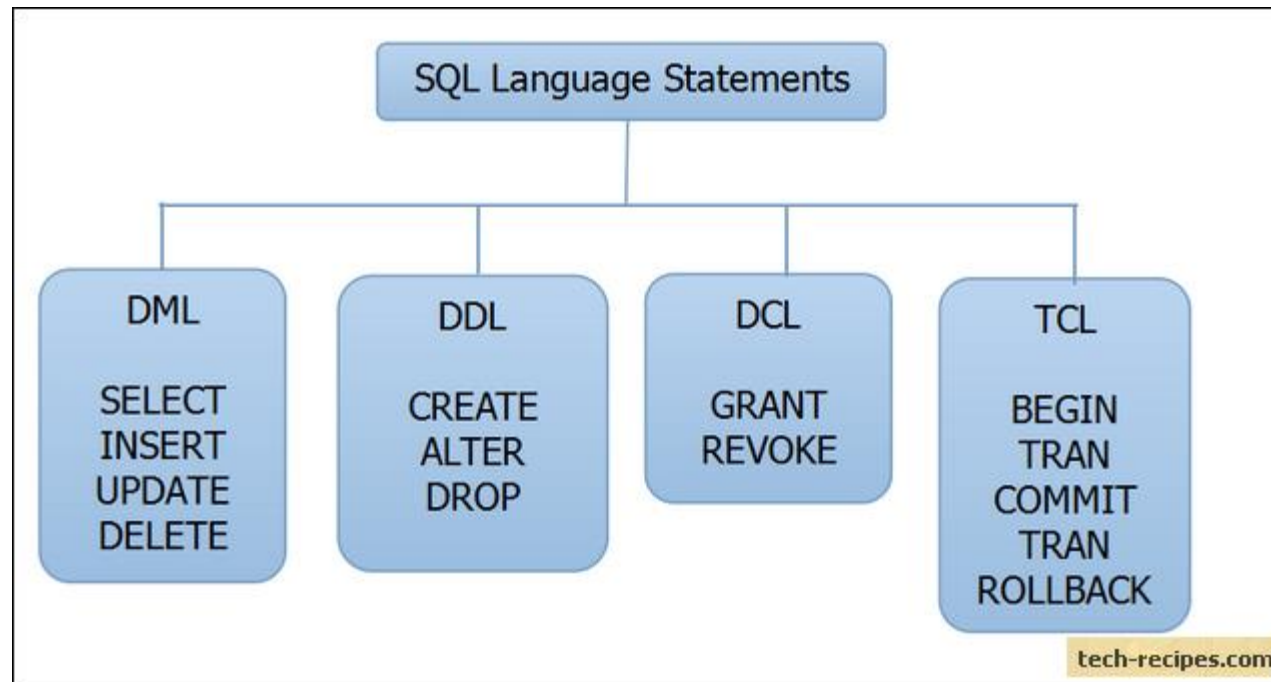
¿Qué debemos esperar para modelar una Base de Datos?

- Los datos.
- La relación que existe entre los datos.
- Restricciones de los datos, se puede frenar error antes que llegue a la capa de aplicaciones.

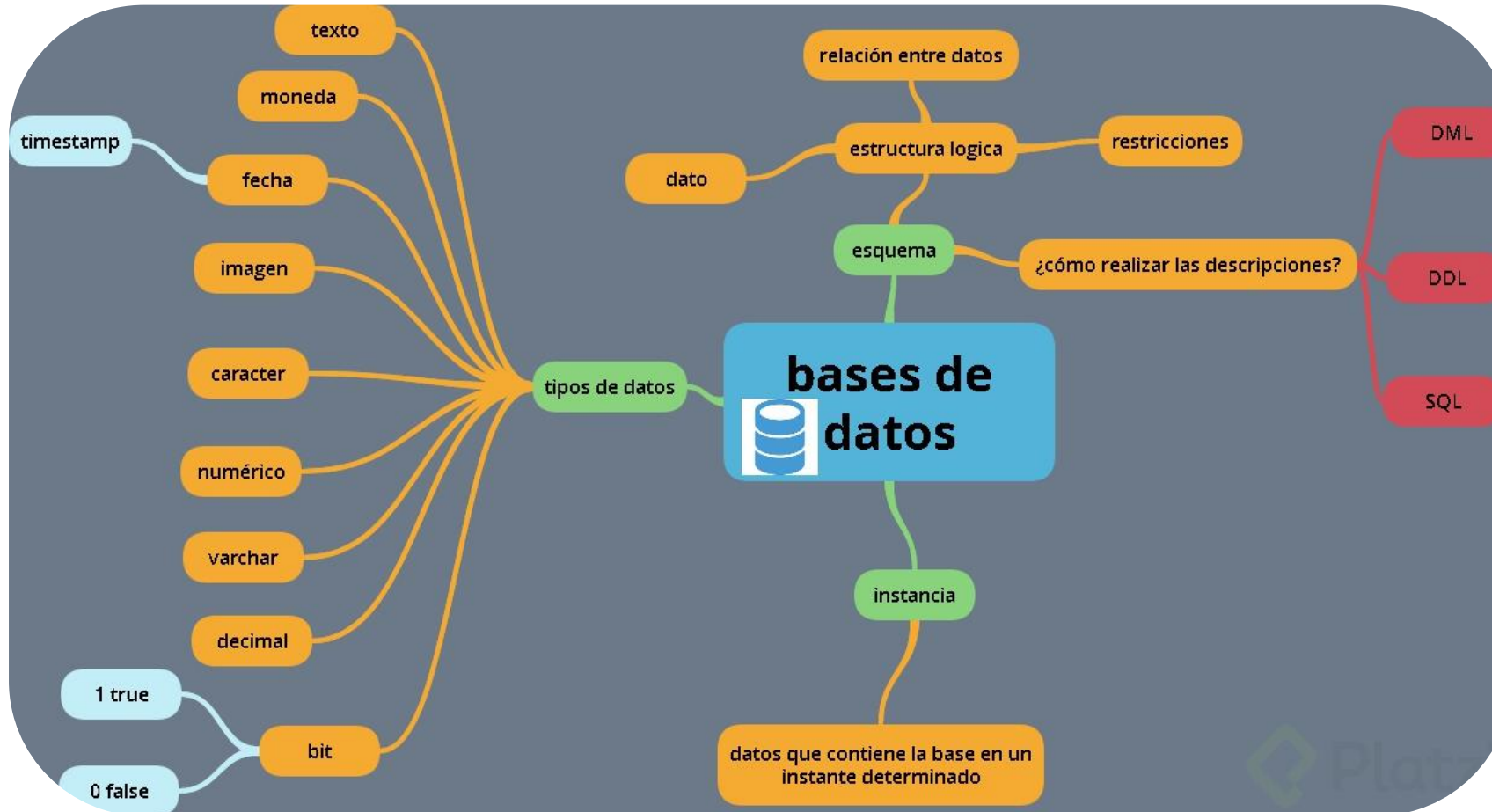
Existen 3 cosas para poder hacer la descripción de una Base de Datos:

- DML = Data Manipulation Language o Lenguaje de Manipulación de Datos (Lenguaje procedimental y declarativo).
- DDL = Data Definition Language o Lenguaje de Definición de Datos (especifica esquema de base de datos, restricciones de integridad, autorizaciones).
- SQL = Structured Query Language o Lenguaje de Consulta Estructurada.

Existen 3 cosas para poder hacer la descripción de una Base de Datos:



TIPO DE DATOS



Características de Bases de Datos SQL:

- Es un lenguaje estructurado.
- Tiene un esquema de tablas.
- Tiene integración con otros tipos de archivos.
- Tiene indexación por medio de árboles.



Bases de Datos No SQL

- La indexación funciona con objetos JSON, no necesariamente funciona como un arbol, se pueden hacer indices dividiendo los objetos por sus caracteristica y particularidades.
- Ejemplos; MongoDB, Cassandra.



Analítica y de Bigdata

- Lenguaje no estructurado.
- Integración de muchos sistemas.
- Sistemas tradicionales y de engagement (social).
- Principio Divide y Venceras.
- Basado en esquemas Scale Out
- Ejemplos hadoop, spark, etc



Bases de datos basadas en aceleración

- Generalmente basadas In Memory (no persisten en el tiempo).
- Uso de aceleradores (Flash Cards, GPU, FPGAs).
- Estructuras diferentes (nodos).
- Son bases de datos muy rápidas que sin embargo no tienen persistencia.
- Ejemplos; Redis, neo4j, Kinetica.

Formas de usos en las bases de datos:

- On premise open source, bases de datos de formato empresarial u opensource instalada en nuestra maquina sin una gran infraestructura.
- Licenciamiento por cores o sockets, se paga dependiendo de ciertas características; como el hardware en el que va a correr.
- Licenciamiento modular, se paga por funcionalidades o modulos para necesidades diferentes.
- Pago por uso a través de SAAS (Software As A Service, ejemplos de SaaS: Google Apps, Salesforce, Workday, Concur, Citrix GoToMeeting, Cisco WebEx.)
 - o PAAS (Platform As A Service, ejem. Apprenda). Es como adquirir una renta y pagar por usar una base de datos.
- Suscripción de nodos de computo, funciona para plataformas como Hadoop el cual no es centralizado y trabaja de forma distribuida, se paga por nodo utilizado.

¿ QUÉ ES UNA ENTIDAD ?



- Una entidad es la representación de un objeto o concepto del mundo real. En las bases de datos se emplea un modelo de datos para describir la estructura de la base de datos.

ENTIDAD: CASA



CARACTERISTICAS :

- Techo
- Puerta
- Ventanas
- Fecha de fabricación
- Color

ENTIDAD : EJEMPLO CONCEPTUAL



ENTIDAD: CASA



CARACTERISTICAS :

- Techo
- Puerta
- Ventanas
- Fecha de fabricación
- Color

CASA

Identificador
Color
Ubicación
Ctd puertas
Cta Ventanas
Fecha de fabricación

Number
Carácter (30)
Texto
Number
Number
Fecha

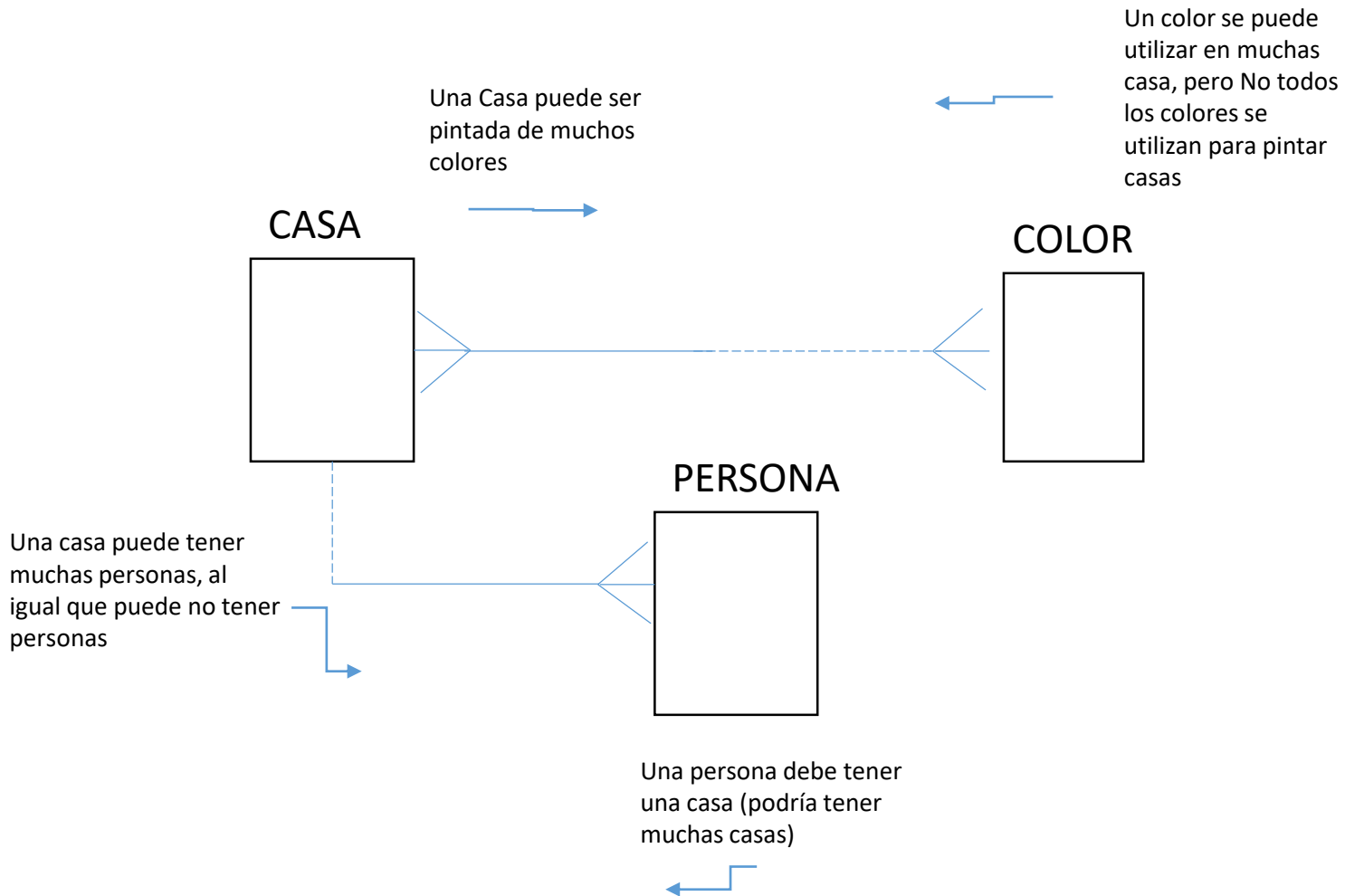
¿ QUÉ ES UNA RELACION ?



Para definir una **Relación** tenemos que tener en cuenta los siguientes puntos:

- La obligatoriedad. Ésta se denota con una línea continua.
- Opcional. Se representa con una línea punteada.
- Datos importantes: El símbolo con el que representamos la característica “de uno a muchos” es con la llamada pata de gallo, que gráficamente es una línea continua con dos líneas en 45 grados en cada lado.

¿ QUÉ ES UNA RELACION ?



NOTACION

| | | |
|-------|---------|--------------------------|
| 0 - 1 | -----> | Relación 0 a 1 |
| 1 - 1 | —————> | Relación 1 a 1 |
| 0 - M | ----->< | Relación 0 a Muchos |
| 1 - M | —————>< | Relación 1 a muchos |
| M - M | >-----< | Relación muchos a muchos |

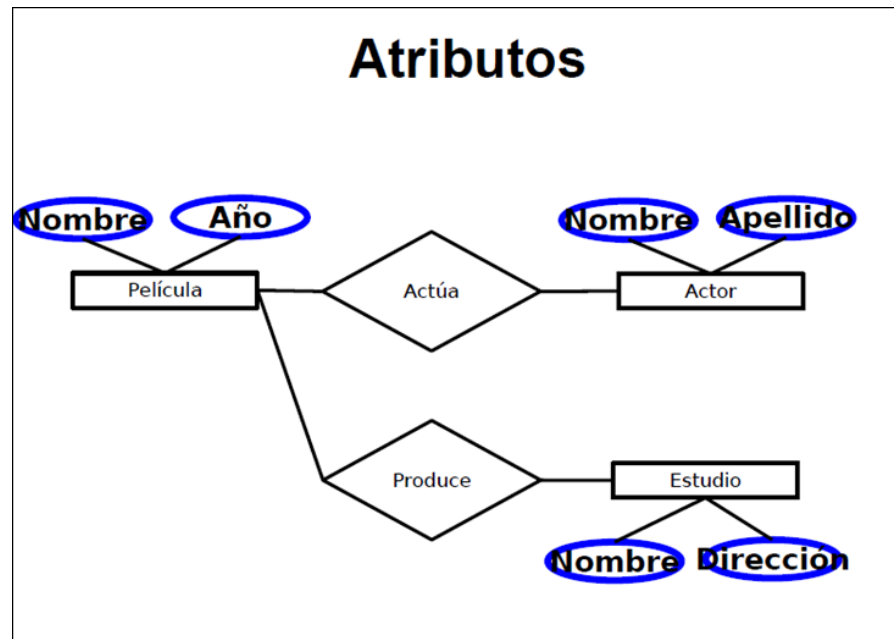


EJEMPLOS

- Con su compañero trabajar un ejemplo de entidades identificando sus relaciones

- Qué es un atributo

Los atributos de una entidad son las características que lo identifican o lo definen



CARACTERÍSTICAS O DATOS DE UNA ENTIDAD



- Se debe tomar en cuenta los tipos de datos que se van a usar en el modelamiento de una entidad.
- Los índices deben ser valores únicos y numéricos (No necesariamente).
- Una entidad puede tener muchas características.

CARACTERÍSTICAS O DATOS DE UNA ENTIDAD



PERSONA

Identificador ID

DNI

Edad

Fecha de nacimiento

Ocupación

Profesión

Color favorito

Teléfono

Number

Carácter (8)

Número (3)

Timestamp

Character (40)

Character (60)

Character (70)

Número (9)

*

*

*

*

0

0

*

*

¿Cuál es obligatorio?

¿ APARECIERON LAS LLAVES ?



- Las llaves nos dan acceso a los datos de una entidad, su notación es la de numeral #, tienen que ser irrepetibles y obligatorias.
- Las llaves son fundamentales por que son obligatoriamente índices, los cuales permiten encontrar los datos cuando se necesitan de una forma rápida y ordenada.
- Una llave primaria de una entidad X, siempre será llave foránea de cualquier otra entidad Y, con la que X se relacione.
- El índice de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla en una base de datos.
- El índice tiene un funcionamiento similar al índice de un libro, guardando parejas de elementos: el elemento que se desea indexar y su posición en la base de datos. Para buscar un elemento que esté indexado, sólo hay que buscar en el índice dicho elemento para, una vez encontrado, devolver un registro que se encuentre en la posición marcada por el índice.

¿ APARECIERON LAS LLAVES ?



PERSONA

Identificador ID

DNI

Edad

Fecha de nacimiento

Ocupación

Profesión

Color favorito

Teléfono

Number

Carácter (8)

Numérico (3)

Timestamp

Character (40)

Character (60)

Character (70)

Numérico (9)

*

*

*

0

0

*

*

¿Cuál es la llave?

¿ APARECIERON LAS LLAVES ?



- Índice único -> restricción que no se puede tener valores repetidos.
- Clave primaria = Índice único ?
- La clave primaria no permite nulo (Not null), el índice único si permite nulos (Null)
- Una entidad puede tener una sola PK, pero puede tener mas de un índice único.
- Tupla / registro / fila -> Representa un objeto único de datos implícitamente estructurados en una tabla, están formados por datos de varias columnas.

- Ejemplo:

| DESPACHOS | | |
|-----------------|---------------|-------------------|
| <i>edificio</i> | <i>número</i> | <i>superficie</i> |
| Marina | 120 | 10 |
| Marina | 122 | 15 |
| Marina | 230 | 20 |
| Diagonal | 120 | 10 |

| EMPLEADOS | | | | | |
|------------|---------------|-----------------|----------------|---------------------|-------------------|
| <i>DNI</i> | <i>nombre</i> | <i>apellido</i> | <i>DNIjefe</i> | <i>edificiodesp</i> | <i>númerodesp</i> |
| 40.444.255 | Juan | García | NULO | Marina | 120 |
| 33.567.711 | Marta | Roca | 40.444.255 | Marina | 120 |
| 55.898.425 | Carlos | Buendía | 40.444.255 | Diagonal | 120 |
| 77.232.144 | Elena | Pla | 40.444.255 | NULO | NULO |

Tupla con referencia incorrecta ->

(55.555.55, maría, Casagran, NULO, París, 400)

- Una clave foránea de una relación R es un subconjunto de atributos del esquema de la relación, que denominamos CF y que cumple las siguientes condiciones:

1) Existe una relación S (S no debe ser necesariamente diferente de R) que tiene por clave primaria CP.

2) Se cumple que, para toda tupla t de la extensión de R, los valores para CF de t son valores nulos o bien valores que coinciden con los valores para CP de alguna tupla s de S.

Y entonces, se dice que la clave foránea CF referencia la clave primaria CP de la relación S, y también que la clave foránea CF referencia la relación S.

¿ APARECIERON LAS LLAVES ?



| ITEM | LLAVE PRIMARIA | LLAVE FORÁNEA |
|-------------------------------------|-----------------|--|
| 1 o más columnas | Sí | Sí |
| Permite duplicados | No | Sí |
| Permite NULL | No | Sí |
| Identifica una fila de manera única | Sí | Quizás |
| Número permitido por tabla | 1 | 1 o más |
| Indexado | Automáticamente | No se crea ningún índice automáticamente |

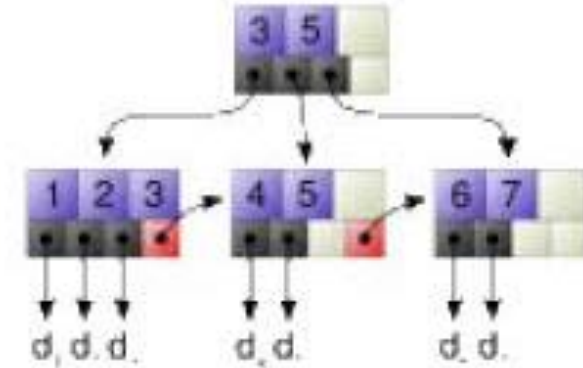
ÍNDICES E INDEXACION

- Las llaves primarias obligatoriamente van a ser índices.
- Las Bases de Datos indexan con un algoritmo llamado: Árboles B+
- Los Árboles B+ son una estructura que va a tener un tronco, tres raíces, de las cuales se van a ir derivando tres raíces más por cada una, hasta donde sea necesario.
- Por defecto todas las Bases de Datos están indexadas, así no le pongamos índices. Lo que sucede es que la Base de Datos siempre obliga a indexar porque siempre tienen un atributo que está oculto, este atributo es RowID.



ROWID :

- Número de datafile donde se almacena la fila
- Dirección del bloque donde está la fila
- Posición dentro del bloque



El acceso mas rápido a una fila es através del ROWID

- Dirección del bloque donde está la fila
- Posición dentro del bloque

CONSTRAINS O RESTRICCIONES



- Las restricciones se pueden trabajar desde la Base de Datos. Normalmente las validaciones con restricciones se hacen desde la aplicación, pero es importante tener en cuenta que podemos hacerlo de igual forma desde la Base de Datos.
- Las llaves primarias y las llaves foraneas no solamente tienen la restricción Not null, sino que además tienen la restricción unique, no puede haber otra igual.
- Con check, las validaciones que podemos hacer son: Igual, mayor o igual, menor o igual, mayor qué o menor qué.

CONSTRAINS O RESTRICCIONES



Cinco tipos de restricciones aplicados en sentencias SQL

| Restricciones (CONSTRAINT) |
|-----------------------------|
| NOT NULL |
| UNIQUE |
| PRIMARY KEY |
| FOREING KEY |
| CHECK |

Ejemplos de restricciones:

- Validación si una persona tiene cierta edad (> 18 años)
- El campo no puede estar vacío Not Null
- El campo no se puede repetir Unique
- Validación por comparación Check ($=$, $>=$, $<=$, $><$).

¿ Que tipos de restricciones han utilizado en base de datos ?

CONSTRAINS O RESTRICCIONES



ALL_CONSTRAINS de Oracle

NOT NULL = campo obligatorio.

```
CREATETABLE Customer  
(SIDintegerNOTNULL,  
Last_Name varchar (30) NOTNULL,  
First_Name varchar(30));
```

UNIQUE = Asegura que todos los valores en una columna sean distinto

```
CREATETABLE Customer  
(SIDintegerUnique,  
Last_Name varchar (30),  
First_Name varchar(30));
```

CHECK = Todos los valores en una columna cumplan ciertas condiciones

```
CREATETABLE Customer  
(SIDintegerCHECK (SID > 0),  
Last_Name varchar (30),  
First_Name varchar(30));
```

CAPAS DE ABSTRACCION DEL MODELO ENTIDAD RELACION

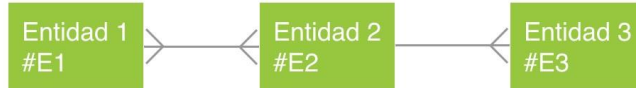


- **Capa Conceptual:** En esta capa vamos a tener varias entidades, aún sin nombre definido. Las entidades van a tener cada una sus llaves primarias y sus atributos, además van a tener relaciones.
- Para que existan las relaciones “muchos a muchos” se necesitan llaves foráneas en las entidades.
- **Capa Lógica:** El modelo Entidad-Relación para poder procesar las relaciones “muchos a muchos” las va a partir en entidades que se llaman: Entidades Débiles.
- **Capa Física:** Este modelo va a ser el paso del modelo lógico hacia la representación que ya va a tener la Base de Datos. En esta capa, ya cada uno de los datos empieza a entrar en las clasificaciones según su tipo de dato.

CAPAS DE ABSTRACCION DEL MODELO ENTIDAD RELACION

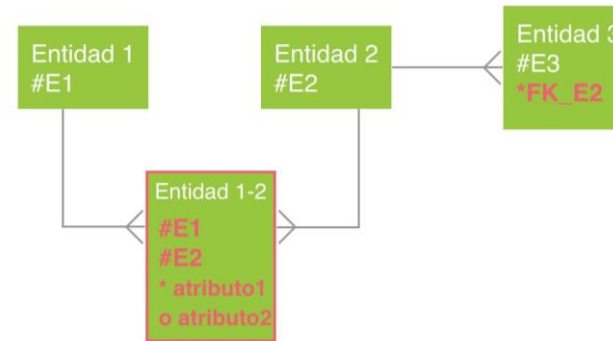


Capa Conceptual



- **Capa Conceptual:** En esta capa vamos a tener varias entidades, aún sin nombre definido. Las entidades van a tener cada una sus llaves primarias y sus atributos, además van a tener relaciones.
- Para que existan las relaciones “muchos a muchos” se necesitan llaves foráneas en las entidades.

Capa Lógica



- **Capa Lógica:** El modelo Entidad-Relación para poder procesar las relaciones “muchos a muchos” las va a partir en entidades que se llaman: Entidades Débiles.

Capa Lógica -> Física



- **Capa Física:** Este modelo va a ser el paso del modelo lógico hacia la representación que ya va a tener la Base de Datos. En esta capa, ya cada uno de los datos empieza a entrar en las clasificaciones según su tipo de dato.

METODOLOGIA DE MODELAMIENTO

PASO 1: IDENTIFICAR ENTIDADES



Ejemplo: Modelar el concepto de Negocio **Aeropuerto**

- **Paso 1:** Vamos a identificar cuáles son las entidades que van a resolver nuestro problema.
- Recomendación: Documentarse muy bien acerca del problema que se va a resolver.

Identificación de entidades
en el ejemplo de
aeropuertos:

- Avión
- Aerolínea
- Ruta
- Tripulante
- Piloto
- Aeropuerto
- País
- Ciudad
- Pasajero

METODOLOGIA DE MODELAMIENTO

PASO 2: IDENTIFICACIÓN DE RELACIONES



Luego de identificar las Entidades, se debe identificar las relaciones de cada entidad.

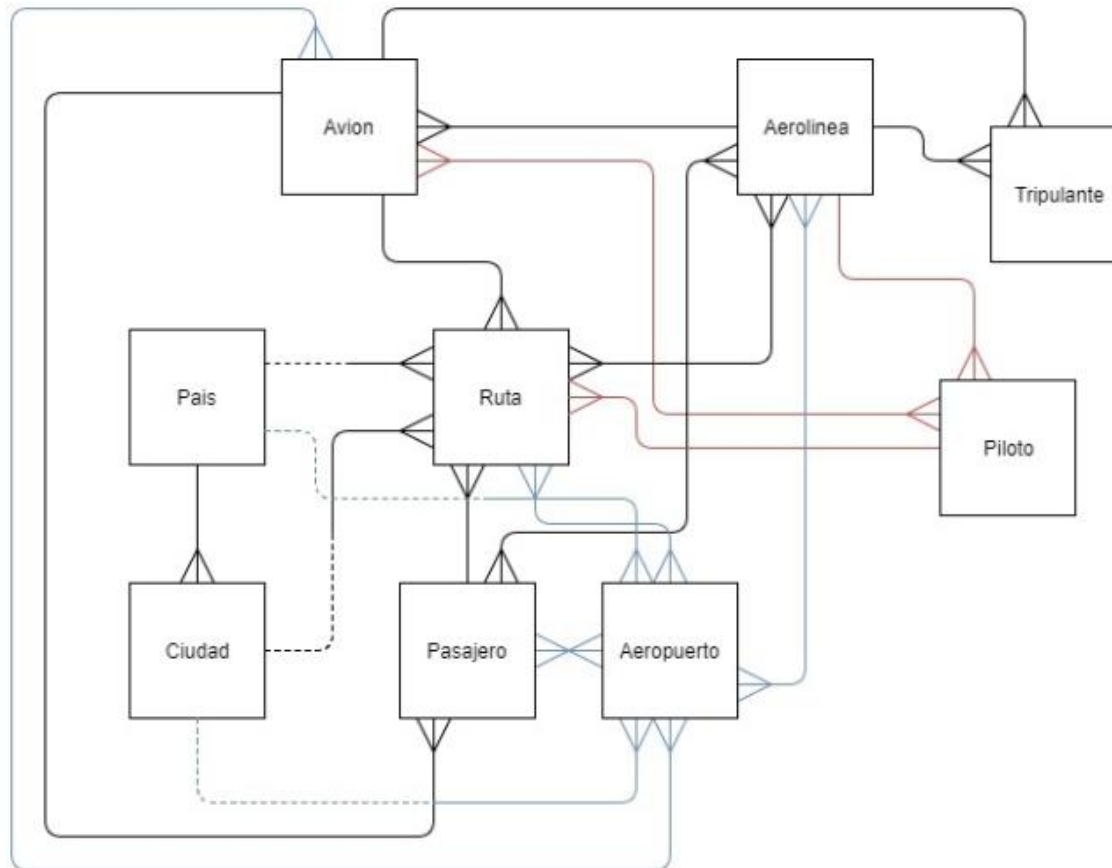
| | Avión | Aerolinea | Ruta | Tripulante | Piloto | Aeropuerto | País | Ciudad | Pasajero |
|------------|-------|-----------|------|------------|--------|------------|------|--------|----------|
| Avión | x | 1:1 | 1:M | 1:M | 1:M | 1:M | x | x | 1:M |
| Aerolinea | 1:M | x | 1:M | 1:M | 1:M | 1:M | x | x | 1:M |
| Ruta | x | 1:M | x | x | x | 1:M | 1:1 | 1:1 | x |
| Tripulante | x | 1:1 | x | x | x | x | x | x | x |
| Piloto | 1:M | 1:1 | 1:M | x | x | x | x | x | x |
| Aeropuerto | 1:M | 1:M | 1:M | x | x | x | 1:1 | 1:1 | 1:M |
| País | x | x | 0:M | x | x | 0:M | x | 1:M | x |
| Ciudad | x | x | 0:M | x | x | 0:M | 1:1 | x | x |
| Pasajero | x | 1:M | 1:M | x | x | 1:M | x | x | x |

METODOLOGIA DE MODELAMIENTO

PASO 3: ENTIDADES Y RELACIONES



Generar el diagrama conceptual.



- Un Avión puede TENER muchas Rutas
- Un Piloto puede ESTAR en una Aerolínea
- Una Aerolínea puede TENER muchos Pilotos

Herramientas de diseño y modelado

- <https://cacoo.com/es/>
- <http://www.vertabelo.com/>
- <https://www.glify.com/>
- <http://dia-installer.de/index.html.es>
- <http://web.balsamiq.com>
- <http://draw.io>
- <http://www.dbwrench.com/>
- <https://dev.mysql.com/>

METODOLOGIA DE DISEÑO CORRECCIONES DEL PASO 2 Y 3



Actualizamos el diagrama conceptual.

| | Avión | Aerolínea | Ruta | Tripulante | Piloto | Aeropuerto | País | Ciudad | Pasajero |
|------------|-------|-----------|------|------------|--------|------------|------|--------|----------|
| Avión | X | 1-1 | X | X | 1-M | 1-M | X | X | X |
| Aerolínea | 1-M | X | 1-M | 1-M | 1-M | 1-M | X | X | 1-M |
| Ruta | X | 1-M | X | X | X | 1-M | X | X | X |
| Tripulante | X | 1-1 | X | X | X | X | X | X | X |
| Piloto | 1-M | 1-1 | X | X | X | X | X | X | X |
| Aeropuerto | 1-M | 1-M | 1-M | X | X | X | X | 1-1 | X |
| País | X | X | X | X | X | X | X | 1-M | X |
| Ciudad | X | X | X | X | X | 0-M | 1-1 | X | X |
| Pasajero | X | 1-M | X | X | X | X | X | X | X |

METODOLOGIA DE DISEÑO

PASO 4: ASIGNAR ATRIBUTOS A ENTIDADES



Para hacer un buen ejercicio hay que pensar en:

- ¿Que atributos voy a necesitar?
- ¿Cual va a ser la codificación que voy a utilizar?
- ¿Como los voy a trabajar dependiendo del tipo de dato?

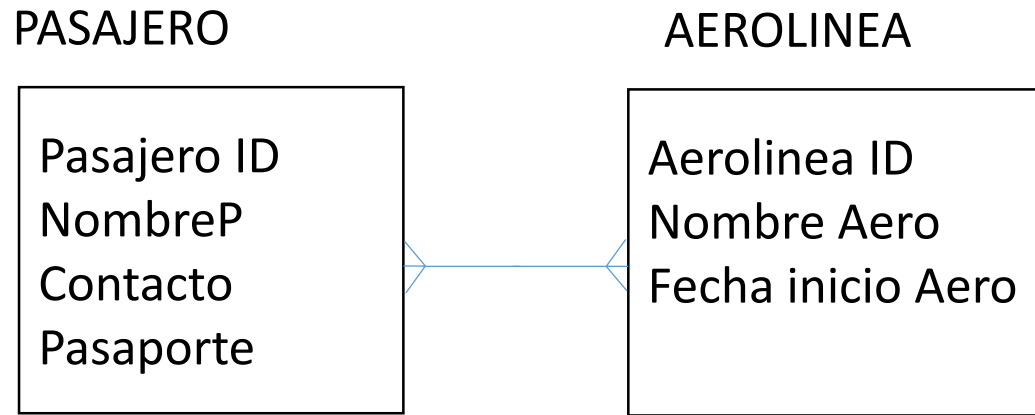
Hasta este punto no nos vamos a fijar en que motor de base de datos vamos a correr.

AVION

| ATRIBUTO | TIPO | OBLIGATORIEDAD | PK |
|--------------|-------------|----------------|----|
| AV_ID | SERIAL | X | X |
| AV_matricula | Varchar(8) | X | X |
| AV_placa | Varchar(6) | X | |
| AV_Nombre | Varchar(15) | X | |
| AV_Tipo | Varchar(20) | X | |
| AV_Marca | Varchar(6) | | |
| | | | |

METODOLOGIA DE DISEÑO

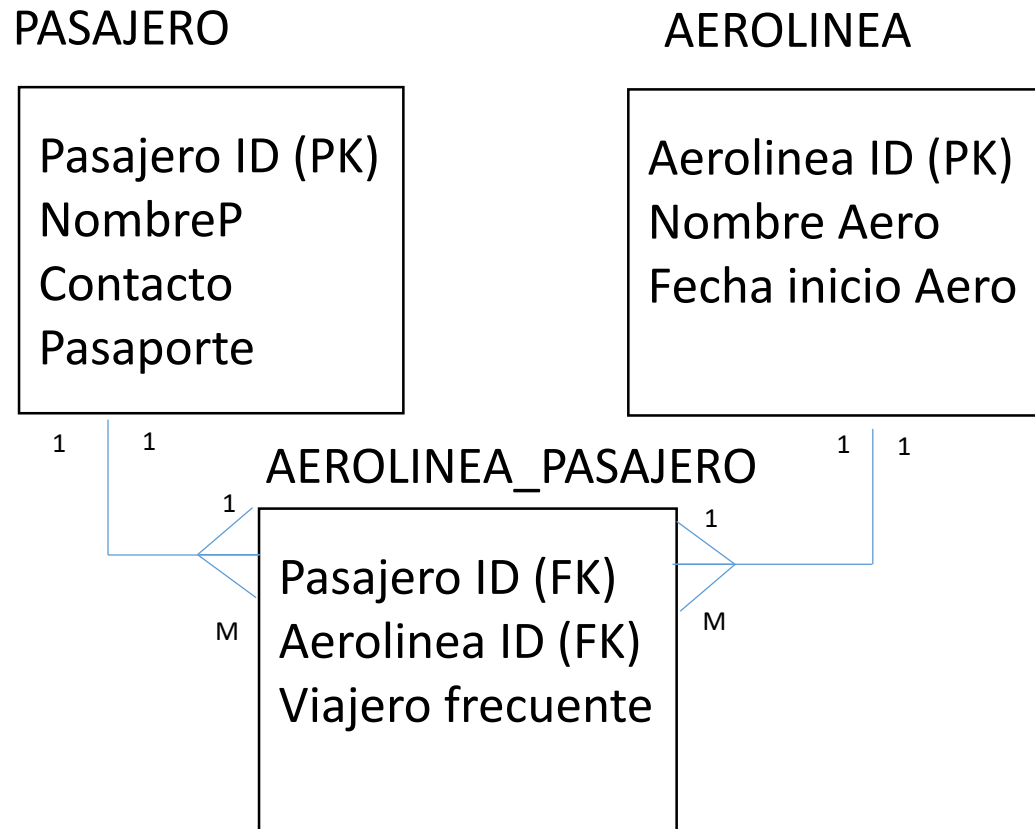
PASO 5,6,7: DIAGRAMA CONCEPTUAL, MODELO LOGICO, NUEVOS ATRIBUTOS A LAS ENTIDADES



- **Paso 5:** Generar un diagrama conceptual (entidades, relaciones y atributos). Las relaciones siempre se van a hacer en 2 sentidos (A con B y B con A).
- **Paso 6:** Modelo lógico, romper relación muchos a muchos, las relaciones se hacen por medio de entidades débiles, entre las entidades relacionadas, esto es porque no podemos generar muchas llaves foráneas en ambas entidades (fuertes).

METODOLOGIA DE DISEÑO

PASO 5,6,7: DIAGRAMA CONCEPTUAL, MODELO LOGICO, NUEVOS ATRIBUTOS A LAS ENTIDADES



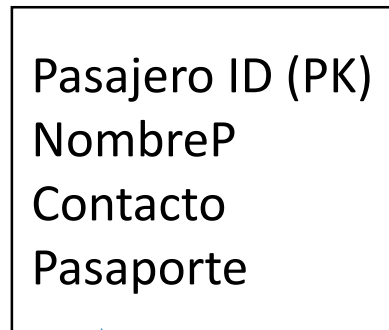
- **Paso 5:** Generar un diagrama conceptual (entidades, relaciones y atributos). Las relaciones siempre se van a hacer en 2 sentidos (A con B y B con A).
- **Paso 6:** Modelo lógico, romper relación muchos a muchos, las relaciones se hacen por medio de entidades débiles, entre las entidades relacionadas, esto es porque no podemos generar muchas llaves foráneas en ambas entidades (fuertes).
- **Paso 7:** Identificar nuevos atributos que generan nuestras entidades débiles.

METODOLOGIA DE DISEÑO

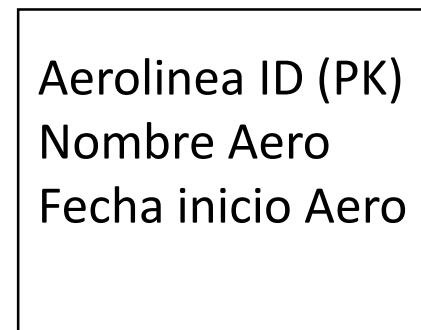
PASO 8: CONSTRUIR DIAGRAMA DEL MODELO FISICO



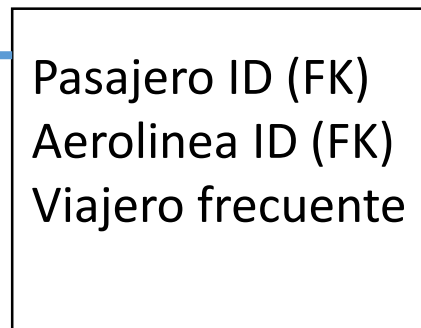
PASAJERO



AEROLINEA



AEROLINEA_PASAJERO



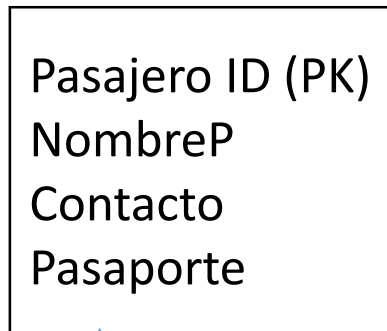
- **Paso 8:** Construir el diagrama del modelo físico
- Las líneas de las relaciones van a ser líneas rectas, la relación uno a muchos vamos a crear una línea con un sentido que termina en cabeza de flecha. Las líneas 1 a 1 o 0 a 1 se mantienen como antes.
- Las llaves foráneas en entidades débiles refieren o apuntan hacia su llave primaria en entidades fuertes

METODOLOGIA DE DISEÑO

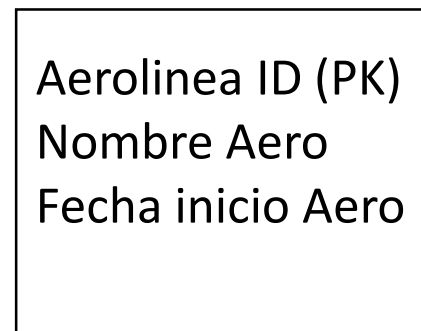
PASO 9: PASAR AL ESTANDAR DE BASE DE DATOS (SQL)



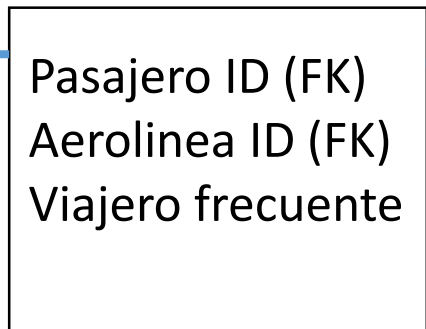
PASAJERO



AEROLINEA



AEROLINEA_PASAJERO



```
CREATETABLE PASAJERO
(
  PASAJERO_ID INTNOTNULL auto_increment,
  NOMBRE_P VARCHAR(30) NOTNULL,
  PASAPORTE VARCHAR(15),
  CONTACTOEMERGENCIA VARCHAR(40) NOTNULL,
  PRIMARY KEY(PASAJERO_ID)
);
CREATETABLE AEROLINEA
(
  AEROLINEA_ID INTNOTNULL auto_increment,
  NOMBRE_A VARCHAR(30) NOTNULL,
  PRIMARY KEY(AEROL_ID)
);
CREATETABLE AROLINEA_PASAJERO
(
  PASAJERO_ID INTNOTNULL,
  AEROLINEA_ID INTNOTNULL,
  VIAJEROFRECUENTE NUMERIC(12,0),
  PRIMARY KEY(PASAJERO_ID, AEROLINEA_ID)
);
ALTERTABLE AEROLINEA_PASAJERO ADDCONSTRAINT FK_AEROLINEA_PASAJERO FOREIGN KEY
(PASAJERO_ID) REFERENCES PASAJERO (PASAJERO_ID);
ALTERTABLE AEROLINEA_PASAJERO ADDCONSTRAINT FK_AEROLINEA_PASAJERO FOREIGN KEY
(AEROLINEA_ID) REFERENCES AEROLINEA (AEROLINEA_ID);
```



Programa Ejecutivo de Business Intelligence

Fundamentos y modelamiento de Base de Datos

Ing. Arturo Rojas Medrano