MODELO JERÁRQUICO

El sistema jerárquico más comúnmente conocido es el sistema IMS de IBM. Esta base de datos tiene como objetivo establecer una jerarquía de fichas, de manera que cada ficha puede contener a su vez listas de otras fichas, y así sucesivamente. P.ej., una ficha de clientes puede contener una lista de fichas de facturas, cada una de las cuales puede contener a su vez una lista de fichas de líneas de detalle que describen los servicios facturados.

Componentes en un Modelo Jerárquico

Una base de datos jerárquica está compuesta por una *secuencia de bases de datos físicas*, de manera que cada base de datos física se compone de todas las *ocurrencias de un tipo de registro o ficha determinada*.

Una *ocurrencia de registro* es una jerarquía de ocurrencias de segmento. Cada ocurrencia de segmento está formada por un *conjunto de ocurrencias o instancias de los campos que componen el segmento*. P.ej., en la figura siguiente tenemos una ocurrencia del tipo de registro Curso, de manera que como cabeza principal tenemos una instancia del segmento curso, de la cual dependen una o varias instancias de los segmentos Requisito y Oferta; a su vez, de Oferta dependen otros que son Profesor y Estudiante.

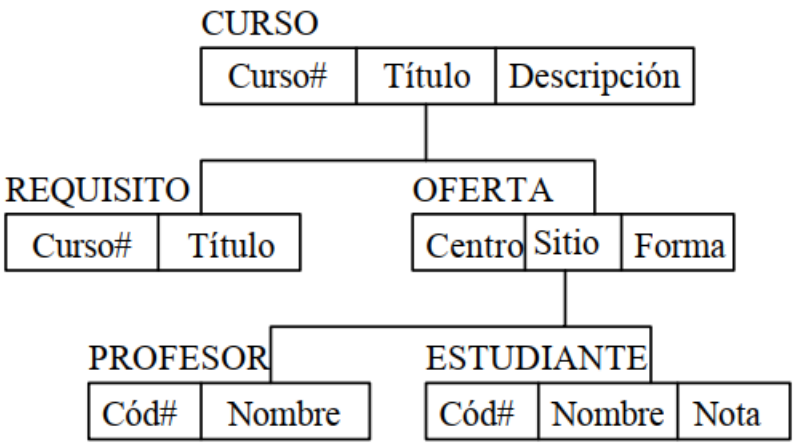

Figura #1. *Ejemplo de tipo de registro.*

Los tipos de segmento son CURSO, REQUISITO, OFERTA, PROFESOR, y ESTUDIANTE. CURSO es el tipo de segmento raíz.

Características en un Modelo Jerárquico

Cabe distinguir en este punto entre el concepto de tipo de registro, y ocurrencia o instancia de registro. El tipo define la estructura general que debe poseer, o sea, los campos de cada uno de sus segmentos, y la estructura jerárquica entre ellos. Una instancia es un valor de un tipo de registro. Para que quede más claro, un tipo de registro es como un tipo de persona: blanco, negro, amarillo, aceitunado, etc., mientras que una instancia es una persona concreta perteneciente a uno de estos tipos: Pablo Picasso, Nelson Mandela, Mao Tse Tung, Toro Sentado, etc.

De esta forma, al segmento que se halla a la cabeza de un registro, se le llama *segmento padre*, y se llama *segmentos hijo* a los que dependen de él. Para movernos por un registro de estructura jerárquica lo que se hace es posicionarse inicialmente en la raíz de una instancia, e ir navegando por sus hijos según nos convenga consultando o modificando los datos pertinentes.

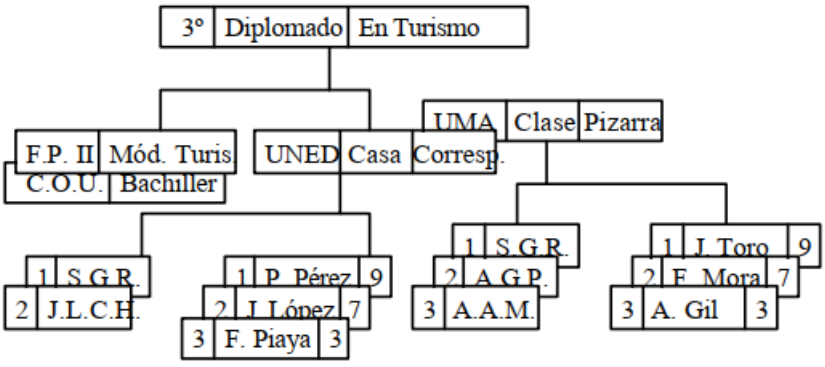

Figura #2. Instancia de un registro

Una base de datos de este tipo no permite el acceso directo a las instancias de un segmento hijo, si no es seleccionando previamente las instancias de los padres de los que depende. P.ej., no se puede seleccionar un estudiante si no es previa

selección de una oferta y de un curso. Las instancias de un mismo segmento que dependen de una misma instancia padre se llaman *instancias gemelas*. en el ejemplo, las instancias:

1	J. Toro	9
2	F. Mora	7
3	A. Gil	3

Tabla #1. Instancias Gemelas. (Figura #2).

Son ocurrencias gemelas, pues todas dependen de la instancia UMA Clase Pizarra del tipo de segmento Oferta. Nótese que si el administrador decide ocultar a determinados usuarios ciertos segmentos (debido a que no tienen por qué tener conocimiento de su existencia), hay que eliminar también todos los segmentos hijos que dependen de él. P.ej., si alguien no debe tener acceso a las ofertas, sólo podrá acceder a los Cursos y a los Requisitos, pero tampoco a los profesores ni a los estudiantes.

No profundizaremos más en este sistema; tan sólo indicar algunos de sus problemas:

- La jerarquía existente entre los tipos de objetos que se manipulan (Cursos, Estudiantes, Profesores, etc.), y las dependencias existentes, hacen que sea imposible el acceso directo a instancias de cada una de ellas, con lo que se pierde en independencia y facilidad de uso.

- Si un mismo segmento debe participar en varios tipos de registro, deben incluirse mecanismos que eviten la repetición de datos. Es más, en el ejemplo anterior se ve que una instancia del segmento Profesor: S.G.R. aparece dependiendo de la oferta de la UNED, y de la UMA. Está claro que los datos no se deben repetir, ya que ello puede provocar que posteriormente se modifique una de las instancias, pero no la otra, con la consiguiente inconsistencia entre ambas copias de los mismos datos.

MODELO DE RED

Podemos considerar al modelo de bases de datos en red como de una potencia intermedia entre el jerárquico y el relacional que estudiaremos más adelante. Su estructura es parecida a la jerárquica, aunque bastante más compleja, con lo que se consiguen evitar, al menos en parte, los problemas de aquél.

Componentes en un Modelo de Red

Los conceptos fundamentales que debe conocer el administrador para definir el esquema de una base de datos de red son los siguientes:

- Registro: Viene a ser como cada una de las fichas almacenadas en un fichero convencional.
- Campos o elementos de datos. Son cada uno de los apartados de que se compone una ficha.
- Conjunto: Es el concepto que permite relacionar entre sí tipos de registro distintos.

Características en un Modelo en Red

Podemos imaginar los registros simplemente como fichas de un fichero. Para ilustrar el concepto de conjunto, supongamos que tenemos un tipo de registro de clientes, y un tipo de registro de vuelos de avión, y supongamos que queremos asociar ambas informaciones, de manera que para cada vuelo queremos saber cuáles son los pasajeros que viajan en él. La forma de hacerlo es a través de un *conjunto*. Un conjunto relaciona dos tipos de registro.

Uno de ellos es el *registro propietario del conjunto*, y el otro es el *miembro*. Veamos el diagrama de la figura siguiente que nos aclarará las cosas un poco más.

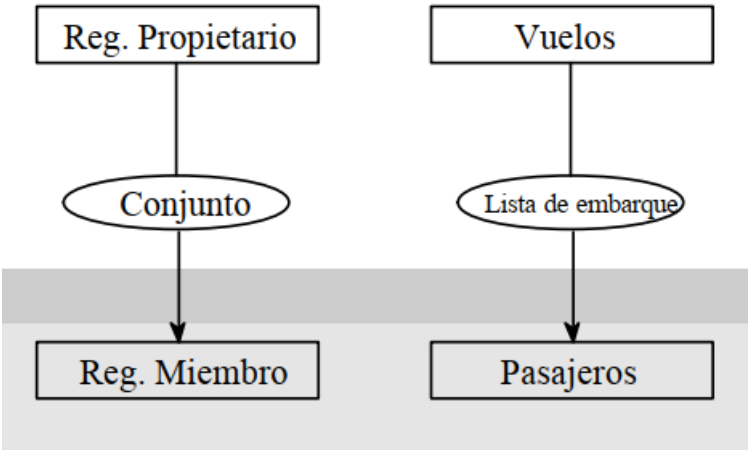

Figura #3. A la izquierda podemos ver el formato general de un conjunto, y a la derecha, el conjunto concreto que nos soluciona saber la lista de embarque de cada vuelo.

Veamos el diagrama de la figura siguiente que nos aclarará las cosas un poco más. Son los *diagramas de Bachman*. Cada tipo de conjunto posee, a su vez, una serie de *ocurrencias de conjunto*, donde cada ocurrencia está formada por una instancia del tipo propietario, y una, varias o ninguna instancia del tipo miembro. P.ej. una ocurrencia de conjunto puede ser:

IB-763 Málaga Helsinki 27/8/97 17:00
33387698-K Juan Linares
83698637-H Pedro Hernández
24885764-G Luis Caro
64653627-J Pablo Mármol

Tabla #2. Ocurrencia de conjunto. (Figura #3).

Una restricción bastante importante de este modelo es que una ocurrencia de registro miembro puede pertenecer como máximo a una sola instancia de un determinado

conjunto, aunque puede participar en varios tipos de conjuntos distintos. Este modelo en red es más potente que el modelo jerárquico, ya que aquél puede simularse, aplicando una jerarquía de conjuntos en varios niveles. P.ej., el ejemplo jerárquico del punto anterior quedaría ahora como:

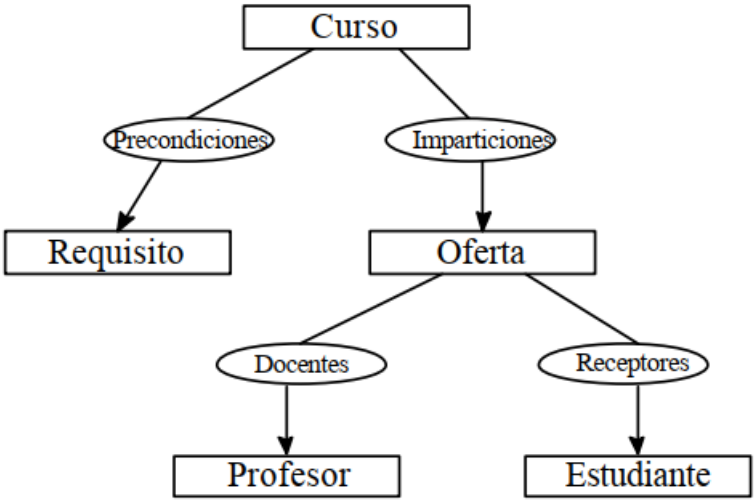

Figura #4. Cómo simular el ejemplo jerárquico mediante el modelo en red.

Por otro lado, en un conjunto concreto, el tipo de registro propietario no puede ser, a su vez, el mismo que el tipo de registro miembro, o sea, un mismo tipo de registro no puede intervenir en el mismo conjunto como propietario y como miembro a la vez. Para ilustrar por qué el modelo en red es más potente que el modelo jerárquico, basta con observar un conjunto como el siguiente:

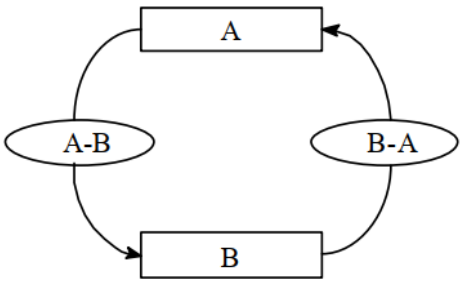

Figura #5. Representación de relaciones entre tipos de registro que no pueden expresarse según el esquema jerárquico.

Que una misma instancia de registro miembro no pueda aparecer en más de una instancia de conjunto, hace que sea difícil de expresar algunas situaciones. P.ej., en el caso de las listas de embarque, está claro que no sólo cada vuelo lo componen varios pasajeros, sino que, además, un mismo pasajero ha podido embarcar en varios vuelos a lo largo de su vida. ¿Cómo representar esta situación? La solución a este problema es algo artificiosa, y pasa por la creación de tipos de registro llamados *enlaces*. La figura siguiente ilustra la solución:

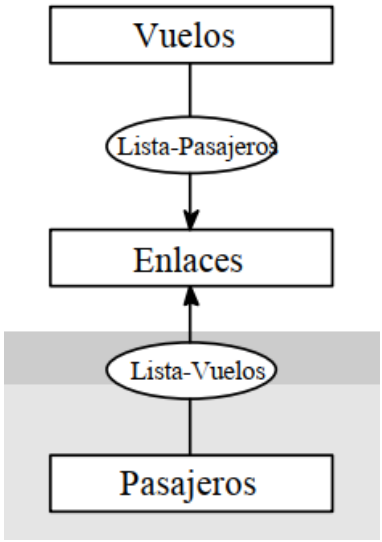

Figura #6. Empleo de enlaces.

Así, cada pasajero se relaciona con una lista de vuelos, que viene dada por una serie de códigos, y cada vuelo se relaciona con una lista de pasajeros que vendrá dada por otra serie de códigos.

MODELO RELACIONAL

Este modelo intenta representar la base de datos como un conjunto de tablas. Aunque las tablas son un concepto simple e intuitivo, existe una correspondencia directa entre el concepto informático de una tabla, y el concepto matemático de relación, lo cual es una gran ventaja, pues permite efectuar formalizaciones de una forma estricta mediante las herramientas matemáticas asociadas, como pueda ser el álgebra relacional en el ámbito de las consultas.

Componentes en un Modelo Relacional

Los conceptos básicos del modelo relacional son:

- Registro: Es algo así como cada ficha de un fichero convencional.
- Tabla: Es un conjunto de fichas de un mismo tipo.

Características en un Modelo Relacional

Con estos dos conceptos anteriores es posible crear cualquier tipo de datos, y asociarlos entre sí, sin las restricciones propias del modelo jerárquico o en red. P.ej., si necesitamos diseñar una base de datos para una agencia de alquiler de coches, necesitaremos una tabla en la que se guarde información sobre los coches, como puede verse en la figura.

Marca	Modelo	Color	Matrícula	Situación
Lamborghi.	Diablo 630	Amarillo	MA-2663-BC	En renta
Ferrari	F-40	Rojo	MA-8870-BC	Disponible
Sbärro R.	Decade	Blanco	VD-870-GTH	Disponible
De Tomaso	Pantera	Blanco	ML-7890-B	En renta
Pontiac	Trans-Am	Negro	KNIGHT	En taller
Austin M.	S3'40	Marrón	CA-5647-AB	Disponible
Jaguar	Destructor	Verde	AD-768-TTY	En renta

Figura #7. Ejemplo de una base de datos relacional.

De esta forma, vemos que cada tabla está compuesta por filas, también llamadas *tuplas o registros*, cada uno de los cuales posee una serie de *campos* en los que se almacenan los datos básicos. El esquema de una tabla nos indica los nombres de cada uno de los campos que contiene, así como el tipo de información que debe contener.

Una tabla es para nosotros un conjunto de registros; por tanto, los registros no pueden repetirse. Para poder acceder a un registro concreto, es necesario hacer una *consulta* a través de algún campo que identifique a dicho registro, como puede ser p.ej. el número de la matrícula. A este campo especial que identifica cada registro se le llama *clave del registro*. La figura siguiente ilustra una tabla de clientes.

Apellidos	Nombre	D.N.I.	Edad
González Aranda	Javier	75836934	27
Beato Apóstol	Antonio	28836746	43
Campos Ortega	Adriano	82665358	36
Ruíz Rojo	Juan	83667228	35

Figura #8. Tabla de clientes de la agencia de alquiler de autos.

En el modelo anterior disponíamos de los conjuntos para asociar información entre sí; ¿cómo nos las apañamos para indicar ahora qué cliente se hace responsable de cada auto alquilado? Fácilmente, a través de una nueva tabla que relaciona los clientes con los autos. Para ello dado que cada registro queda identificado por su clave, nos basta con incluir en esta nueva tabla a las claves de ambas tablas, en lugar de todos sus campos. Así, podemos obtener una nueva tabla de alquileres que contenga la matrícula del auto, y el D.N.I. (identificación personal) del cliente, tal como se ve en la figura siguiente.

Matrícula	D.N.I.
MA-2663-BC	75836934
ML-7890-B	83667228
AD-768-TTY	75836934
AD-768-TTY	82665358

Figura #9. Tabla que relaciona los autos en renta con los clientes que se responsabilizan de ellos.

En esta última tabla podemos observar varias cosas interesantes. Por un lado, un cliente se puede responsabilizar de más de un auto, o sea, puede alquilar más de un auto, pues vemos que Javier González Aranda ha alquilado tanto el Lamborghini como el Jaguar. Pero, a su vez, más de una persona puede hacerse cargo de un auto: Javier González Aranda y Adriano Campos Ortega comparten el alquiler del Jaguar.

De esta forma, el modelo relacional soluciona el problema que se planteaba en el caso de las listas de embarque mediante enlaces artificiosos, y lo soluciona de una manera intuitiva a través de las tablas, y eliminando el concepto de conjunto.

Este método de expresar los datos facilita además las consultas, que se realizan ahora a través de estas tablas especiales que relacionan a otras tablas. P.ej., si queremos saber los autos que ha alquilado González Aranda, basta con buscar su clave en la tabla de clientes (75836934), y a continuación ver que matrículas tiene asociadas en la tabla de alquileres (MA-2663-BC, y AD-768-TTY); a continuación, buscamos en la tabla de autos cuales son los autos que poseen esas claves, y obtenemos como resultado: Lamborghini y Jaguar.

MODELO NO RELACIONAL

Las bases de datos asociadas a este modelo reciben el nombre de NoSQL, siglas que significan “Not Only SQL” se refiere a un grupo de sistemas de bases de datos no relacionales; cuya idiosincrasia principal es que no se encuentran construidas en tablas y generalmente no se usa lenguajes comunes del SQL para manipular los datos. Los sistemas de bases de datos en NoSQL también por su naturaleza tienen la utilidad en la facilidad de trabajar con grandes cantidades de datos, uno de los requerimientos necesarios para la analítica de datos propuesto por el Big data.

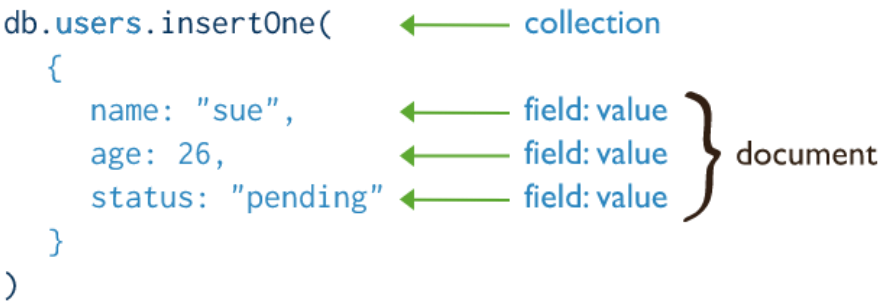

Figura #10. Ejemplo de instrucciones en una base de datos NoSQL (documental).

Características Generales en el Modelo No Relacional

Los sistemas NoSQL se encuentran diseñados para el almacenamiento de datos en larga escala y procesamiento de datos en paralelo a través de varios servidores, son usados por varias compañías reconocidas en internet como Google, Amazon y Facebook, cuyos retos yacían en almacenamiento de grandes cantidades de datos los cuales las RDBMS (Relational Data Base Management Systems) no podían soportar debido a sus diseños. Los sistemas NoSQL pueden soportar múltiples actividades de consulta y análisis de tipo predictivo como exploratorio. Los sistemas NoSQL se encuentran diseñadas de acuerdo por los parámetros establecidos por el ACID (Atomicidad Consistencia Aislamiento Durabilidad), BASE (Basic Availavility Soft state Eventual consistency), OLTP (Online Transaction Processing) en tiempo real y la solución OLAP (On-Line Analytical Processing) garantizando la necesidad de las organizaciones en sistemas que puedan almacenar y trabajar datos en cantidades descomunales.

Tipos de Bases de Datos No Relacional

- Clave-valor: las bases de datos clave-valor son altamente divisibles y permiten escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar. Los casos de uso como juegos, tecnología publicitaria e IoT se prestan particularmente bien con el modelo de datos clave-valor.

- Documentos: en el código de aplicación, los datos se representan a menudo como un objeto o un documento de tipo *JSON* porque es un modelo de datos eficiente e intuitivo para los desarrolladores. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el uso del mismo formato de modelo de documento que emplean en el código de aplicación. La naturaleza flexible, semiestructurada y jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las

aplicaciones. El modelo de documentos funciona bien con catálogos, perfiles de usuario y sistemas de administración de contenido en los que cada documento es único y evoluciona con el tiempo.

- Gráficos: el propósito de una base de datos de gráficos es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de gráficos incluyen redes sociales, motores de recomendaciones, detección de fraude y gráficos de conocimiento.

- En memoria: las aplicaciones de juegos y tecnología publicitaria tienen casos de uso como tablas de clasificación, tiendas de sesión y análisis en tiempo real que requieren tiempos de respuesta de microsegundos y pueden tener grandes picos de tráfico en cualquier momento.

CONCLUSIONES

El avance constante de la tecnología condiciona la practicidad de los modelos de bases de datos tradicionales, por ende, partiendo desde la base de garantizar la disponibilidad y seguridad de los datos asociados, se ha de llegar a un modelo que unifique características en función de los tiempos cambiantes, quizá sea un ideal, o quizá ya lo tengamos entre nosotros, la cuestión reside en saber adaptarnos y estar en constante formación para si ser profesionales capaces de aportar valor a través de un conocimiento actualizado y significativo.

REFERENCIAS BIBLIOGRÁFICAS

VTU e-Learning Centre. (2003). Retrieved 27 May 2021, from https://elearningatria.files.wordpress.com/2013/10/introduction.pdf

ANEXO

SQL en comparación con Terminología NoSQL

La siguiente tabla compara la terminología utilizada por las bases de datos NoSQL seleccionadas con la terminología utilizada por las bases de datos SQL.

SQL	MongoDB	DynamoDB	Cassandra	Couchbase
Tabla	Conjunto	Tabla	Tabla	Bucket de datos
Fila	Documento	Elemento	Fila	Documento
Columna	Campo	Atributo	Columna	Campo
Clave principal	ObjectId	Clave principal	Clave principal	ID del documento
Índice	Índice	Índice secundario	Índice	Índice
Ver	Ver	Índice secundario global	Vista materializada	Ver
Tabla u objeto anidado	Documento incrustado	Mapa	Mapa	Mapa
Matriz	Matriz	Lista	Lista	Lista

Fuente: https://aws.amazon.com/es/nosql/