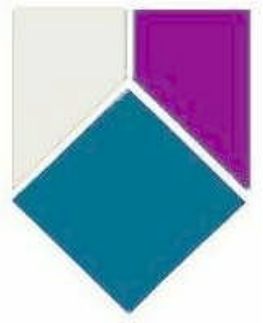
El modelo relacional de bases de datos



Javier Quiroz

El contenido entero de una base de datos relacional se representa por una y sola una forma, a saber: como valores de atributos en tuplas dentro de relaciones.

E. F. Codd

A su memoria: *honor a quien honor merece…*

El 18 de abril del 2003 falleció el Dr. Edgar Frank Codd, a la edad de 79 años víctima de un ataque al corazón. Aún si usted nunca escuchó o supo del Dr. Codd, lo más probable que a diario utilice cotidianamente tecnología derivada de las teorías de este brillante matemático y científico de la computación. Nacido en Inglaterra, la mayor parte de su vida la pasó en los Estados Unidos trabajando y desarrollando sus ideas que culminaron en una serie de informes técnicos acerca de una nueva manera de organizar y acceder los datos. A partir de estos trabajos publicó en 1970 el artículo *A Relational Model of Data for Large Shared Data Banks*, algo así como *Un modelo de datos relacional para grandes bancos de datos compartidos.*

Codd propuso que los sistemas de bases de datos deberían presentarse a los usuarios con una visión de los datos organizados en estructuras llamadas *relaciones*, definidas como conjuntos de tuplas (filas) y no como series o secuencias de objetos, con lo que el orden no es importante. Por tanto, detrás de una relación puede haber cualquier estructura de datos compleja que permita una respuesta rápida a una variedad de consultas. Codd hizo entonces

El autor es Coordinador de Sistemas de Información y Bases de Datos, Dirección de Informática de la Dirección General de Estadística, INEGI.

énfasis en que el usuario de un sistema relacional sólo debía preocuparse por el qué consultar y no el cómo de las estructuras de almacenamiento (lo que ahora se conoce como *modelo físico*). Aún hoy se consideran validas sus afirmaciones, especialmente:

"Los usuarios futuros de grandes bancos de datos deben ser protegidos de tener que saber cómo están organizados los datos en la máquina (la representación interna. […] Las actividades de los usuarios en sus terminales y la mayoría de programas de aplicación no debería verse afectados cuando se cambia la representación interna de los datos o incluso cuando se cambian algunos aspectos de la representación externa. Se necesitará cambiar la representación de los datos a menudo como resultado de los cambios en el tráfico de las consultas, actualizaciones e informes y como consecuencia del crecimiento natural en los tipos de información almacenada."

Puede parecer extraño, pero las ideas de Codd no fueron "recibidas con los brazos abiertos" en IBM, donde realizaba sus labores de investigación, según afirma Harlwood Kolsky, un físico y antiguo compañero de Codd; "fue un enfoque revolucionario", recuerda Kolsky. El nuevo enfoque de Codd, basado en la teoría matemática de conjuntos, no tuvo eco inmediato en IBM, que prefirió a IMS, un producto al que se le había invertido una fuerte cantidad de esfuerzo y dinero.

Un grupo de la Universidad de Berkeley en California, lidereado por Michael Stonebreaker, creyó en la idea del modelo relacional y obtuvo financiamiento para desarrollar un sistema, el *Ingres*, cuya primera versión se presentó en 1974 y fue el primer manejador relacional de bases de datos funcional. Esto tuvo como consecuencia que IBM reaccionara poniendo en marcha otro sistema relacional, el *System R* con características de multiusuario y un lenguaje de consulta estructurado, el *SEQUEL* que luego pasaría a llamarse SQL (*Structured Query Language*). Para entonces Larry Ellison, un empresario del Valle del Silicón, había tomado ventajas de los escritos de Codd para crear un nuevo producto y una nueva empresa que hasta la fecha se conoce como *Oracle*.

En 1985 Codd publicó sus famosas 12 reglas sobre el modelo relacional de bases de datos, un resumen de sus características fundamentales. Es preciso resaltar que todavía hoy algunas de estas reglas son de difícil implementación para los fabricantes de manejadores de bases de datos relacionales. Además de ser considerado como el padre del modelo relacional, Codd también incursionó en el modelo multidimensional de análisis de datos conocido como OLAP (*On Line Analytical Processing*) y en 1993 Codd y algunos de sus colegas publicaron las "12 reglas para OLAP".

A lo largo de su vida, el Dr. Codd recibió innumerables reconocimientos. En 1981, la ACM *(Association for Computer Machinery)*, otorgó a Codd el "Premio Turing", considerado uno de los más prestigiosos en el campo de la informática. Muchos de sus compañeros y seguidores han contribuido, y siguen haciéndolo, a fortalecer el modelo el cual es, por mucho, el más utilizado actualmente como sistema de bases de datos.

La esencia del modelo

La estructura fundamental del modelo relacional es la *relación*, es decir una tabla bidimensional constituida por filas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad. Por ejemplo, si en la base de datos se tienen que representar personas, podrá definirse una relación llamada "Personas", cuyos atributos describen las características de las personas. Cada tupla de la relación "Personas" representará una persona concreta. Por ejemplo, la relación:

Personas (RFC, nombre, apellido, sexo, estadoCivil, fechaNacimiento)

es apenas una definición de la estructura de la tabla, es decir su nombre y la lista de atributos que la componen. Si esta estructura se puebla con datos, entonces tendremos una lista de valores individuales para cada tupla, atributo por atributo.

Aunque una relación es más conocida como *tabla*, las tuplas como *filas* y los atributos como *columnas*, en este escrito usaremos la terminología original[18] y de donde deriva el nombre del modelo. Las tuplas en una relación son un conjunto en el sentido matemático del término, es decir una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de "llave primaria", o sea un atributo o conjunto de atributos que permiten identificar unívocamente una tupla en una relación (en el ejemplo, el atributo RFC cumple con esta función). Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla ("llaves candidatas"), pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo (que significa un valor no determinado), en tanto que ya no permitirían identificar una tupla concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias se conoce como integridad de las entidades.

Cada atributo de una relación se caracteriza por un nombre y por un dominio. El dominio indica qué valores pueden ser asumidos por una columna de la relación. A menudo un dominio se define a través de la declaración de un tipo para el atributo (por ejemplo diciendo que es una cadena de diez caracteres), pero también es posible definir dominios más complejos y precisos. Por ejemplo, para el atributo "sexo" de nuestra relación "Personas" podemos definir un dominio por el cual los únicos valores válidos son 'M' y 'F'; o bien para el atributo "fechaNacimiento" podremos definir un dominio por el que se consideren válidas sólo las fechas de nacimiento después del uno de enero de 1960, si en nuestra base de datos no está previsto que haya personas con fecha de nacimiento anterior a

[18] No confundir *relación* con el mismo término usado en el modelado de Entidad-Relación que se usa para describir las asociaciones que existen entre entidades.

esa. El motor de datos se ocupará de controlar que en los atributos de las relaciones se incluyan sólo los valores permitidos por sus dominios. Característica fundamental de los dominios de una base de datos relacional es que sean "atómicos", es decir que los valores contenidos en los atributos no se puedan separar en valores de dominios más simples. Más formalmente se dice que no es posible tener atributos con valores múltiples (multivaluados).

La *normalización*, o sea la razón y uso de las formas normales, es evitar la repetición innecesaria de datos (redundancia). Una solución a este problema es repartirlos en varias relaciones y utilizar referencias por valor entre ellas. Este es un ejemplo típico de que la tupla de una relación, digamos de Empleados, no deba repetir toda la información de su departamento, sino que debe utilizar una referencia por valor a la tupla de la relación Departamento, donde están todos estos datos. Este procedimiento ahorra espacio de almacenamiento, optimiza el rendimiento y, al eliminar la redundancia, impide modificaciones parciales o incompletas que podrían dar lugar a inconsistencias. Existen hasta 6 formas normales pero, en la práctica, se adopta generalmente la tercera forma normal.

Junto con el modelo, el Dr. Codd también propuso el álgebra relacional, un lenguaje formal con una serie de operadores que trabajan sobre una o varias relaciones para obtener otra relación resultado, sin que cambien las relaciones originales. Tanto los operandos como los resultados son relaciones, por lo que la salida de una operación puede ser la entrada de otra operación. Esto permite anidar expresiones del álgebra del mismo modo que se pueden anidar las expresiones aritméticas. Codd originalmente propuso ocho operandos pero sólo cinco son fundamentales: *restricción, proyección, producto cartesiano*, *unión* y *diferencia*, que permiten realizar la mayoría de las operaciones de obtención de datos. Los operadores no fundamentales son la *concatenación (join)*, la *intersección* y la *división*, que se pueden expresar a partir de los cinco operadores fundamentales. La restricción y la proyección son operaciones *unarias* porque operan sobre una sola relación. El resto de las operaciones son *binarias* porque trabajan sobre pares de relaciones.

Partiendo del cálculo relacional, el Dr. Codd desarrolló el primer lenguaje relacional llamado ALPHA el cual formó el fundamento para el desarrollo subsecuente de lenguaje SQL (original SEQUEL). Otros lenguajes relacionales de consulta, tales como el QBE[19], se basaron en el álgebra relacional definida por el Dr. Codd.

Durante las fases de desarrollo del modelo relacional, el comité ANSI/SPARC de 1975 definió la separación en tres niveles de los sistemas manejadores de bases de datos: externo, conceptual e interno que vinieron a redundar en lo que ahora se conoce como *subesquema externo, esquema lógico y esquema físico*. En otras palabras: los modelos conceptual, lógico y físico. Sin embargo, fue el Dr. Codd quien estableció los fundamentos para esta separación con conceptos tales como la *independencia lógica y física de los datos* (reglas 8 y 9), de *independencia, integridad y distribución* (reglas 10 y 11). Como consecuencia, el

19 Acrónimo para *Query by Example.*

mundo de las bases de datos cambió para siempre. A partir del modelo relacional el usuario no tendría porqué preocuparse de los aspectos técnicos de la base de datos: simplemente sus necesidades de información se satisfarían de acuerdo a su perspectiva de los datos. Igualmente, los programadores de aplicaciones no tendrían que lidiar con el modelo físico, asunto exclusivo del administrador de bases de datos (ABD) quien, si así lo determinara conveniente en aras de un mejor rendimiento, podría modificar el modelo físico de datos sin afectar al modelo lógico. Como veremos, muchos de estos conceptos fundamentales aún son confundidos o ignorados por analistas, desarrolladores y fabricantes que aún no han entendido o podido implementar un verdadero modelo relacional de bases de datos.

El debate Entidad-Relación Vs. esquema en estrella

En la medida que el modelo relacional ganó aceptación en la industria surgieron nuevas ideas y propuestas. En la década de 1990 hubo un fuerte impulso en pro de la tecnología del *data warehousing*[20]. Como ocurre frecuentemente con la adopción de nuevos paradigmas, se registró también un considerable índice de fracasos. La cuestión fue ¿cómo implementar un *data warehouse?* De las soluciones emprendidas resaltaron a) el modelado Entidad-Relación (E-R)[21] donde las entidades representan relaciones normalizadas (por lo general, en tercera forma normal); y b) el *esquema en estrella* (E-E), donde las entidades se modelan en tablas con *hechos* y *dimensiones.*

La regla para una relación normalizada dice: *todos los atributos no llave de una relación dependen sólo y exclusivamente de la llave.* Por definición, el atributo o atributos que componen la llave no tienen valores duplicados; en consecuencia, los atributos no-llave, dependientes por completo de la llave, tampoco. *Debido a esta dependencia* no existe redundancia en las relaciones. Así, en una base de datos completamente normalizada, es suficiente para el diseñador declarar restricciones al manejador sustentadas en la llave, *lo cual garantiza la integridad y consistencia de la base de datos.*

Por contraste, el esquema en estrella se fundamenta en una tabla[22] de *hechos* que contiene uno o más datos que en lo posible deben ser *aditivos* o mensurables y una o más tablas de *dimensiones.* La llave primaria de la tabla de hechos es una concatenación de las llaves primarias de cada una de las dimensiones. Los *hechos,* con frecuencia datos agregados, pueden pensarse como medidas tomadas de la intersección de todas las dimensiones. Los atributos en las dimensiones son, por lo general, textuales y discretos, con

---

[20] A principios de la década de 1980, W. H. Inmon acuñó el término *data warehouse* como una "*colección de datos integrada, temática e histórica, siempre disponible y con variaciones en el tiempo, diseñada para soportar la gestión en la toma de decisiones.* "

[21] Modelo de datos que describe las entidades, atributos y asociaciones de una base de datos

[22] Preferimos llamar "tabla" y no "relación" a las entidades del E-E puesto que, al ser desnormalizadas, no cumplen con el modelo relacional.

el propósito de establecer restricciones específicas en las consultas. A partir del esquema en estrella pueden derivarse "cubos"[23] para su análisis con herramientas OLAP[24]. La construcción de una base de datos bajo este esquema requiere de una rígida definición de hechos y dimensiones (muchas veces sin atender las reglas de normalización) que anticipen la consulta de datos bajo patrones preestablecidos. Supuestamente bajo el E-E se logran mejores tiempos de respuesta y los usuarios "comprenden" mejor los alcances del modelo.

El principal problema con el E-E es la rigidez de su diseño que no permite el descubrimiento de nuevas relaciones: cualquier visión de los datos no prevista es imposible; además, el usuario debe conocer todas las dimensiones de los datos que desea explorar. Si son requeridos otros hechos y dimensiones la "estrella"[25] tiene que ser remodelada y recargada o generarse una nueva. Esto es consumidor de tiempo y recursos, inhibe el descubrimiento de nuevos patrones e información que pueden ser relevantes para la organización (minería de datos). Más aún, si hechos y dimensiones son desnormalizados (grupos repetitivos) estamos ante una aberración del modelo relacional al permitir redundancia y en riesgo de perder la integridad de la base de datos puesto que las restricciones sobre las llaves en las relaciones no tienen validez para mantenerla. En este contexto, el manejador es incapaz de garantizar la integridad. Y si no podemos mantener la integridad y consistencia de la base de datos en un E-E *¿de qué manera podemos afirmar que los datos almacenados son correctos?* Una peligrosa aproximación a las consecuencias de la Ley de Murphy: *"si se permite que ocurra, ocurrirá"*. Por esta razón los diseñadores de un E-E, en el mejor de los casos y cuando lo hacen, tienen que construir software de mantenimiento para cada base de datos en E-E a fin de contar con un medio para demostrar su exactitud, una labor onerosa y no exenta de riesgos perfectamente evitable con otro tipo de soluciones.

El modelado E-R, modela los datos de acuerdo a la manera en que se asocian unos con otros y, conforme crecen los desafíos de la organización, los datos no necesitan reacondicionarse para nuevas y desconocidas relaciones. Aquí tenemos la flexibilidad y belleza del auténtico modelo relacional. El modelado E-R proporciona el sustento para la naturaleza transaccional de cualquier *data warehouse* permitiendo el verdadero análisis exploratorio ejemplificado por la minería de datos. Un modelo E-R es para toda la organización y cualquier relación posible dentro de la misma también es posible en el modelo. Es oportuno enfatizar que un *data warehouse* abarca desde el proceso de recolección, administración y distribución de los datos, no sólo visiones agregadas y predefinidas. Por lo tanto, los sistemas de información deben sustentarse en un ambiente capaz y flexible para todo tipo de consultas, sea para el análisis complejo como a nivel

---

[23] Vista "multidimensional" de los datos. P. ej. una figura de ventas con perspectiva simultánea del tiempo, región, producto, etc.

[24] Acrónimo para *On Line Analytical Processing.*

[25] Los modeladores del E-E ubican, por lo general, la tabla de hechos al centro del dibujo y las tablas de dimensiones a su alrededor. Según esto, el dibujo se parece a una estrella.

detalle y agregado. En este escenario, simple y sencillamente el E-E no soporta los requerimientos a largo plazo de la organización como un todo. Un *data warehouse* sustentado en el E-E no permite al usuario expandir su entendimiento de las asociaciones entre los datos, cambiar la forma en que se visualiza la organización, penetrar en análisis más avanzados o la generación de tipos de datos emergentes.

Se dice que el modelado E-R es muy complejo como para que los usuarios lo entiendan suponiendo, por contraste, que el E-E es fácil de entender. El asunto aquí es que las capas lógica y física de cualquier modelado de bases de datos *no tiene el usuario porqué conocerlas y, en la mayoría de los casos, no debe*. El acceso a una base de datos, por ejemplo la consulta, debe dársele al usuario en función de metadatos, herramientas profesionales y aplicaciones a la medida. Dicho de otra forma, *en sus propios términos y respetando las reglas del negocio.* El hacer partícipe obligado al usuario de la jerga técnica (e. g., *cubos, hipercubos, hechos, dimensiones, relaciones, asociaciones, llaves, etc.)* es desviar a la informática de su función principal, un ejercicio de petulante auto complacencia que no valdría la pena siquiera mencionar de no ser por su negativo impacto en la relación con el usuario final y, en última instancia, en la misión y visión organizacionales.

Otro argumento a favor del E-E es que "las tablas desnormalizadas en hechos y dimensiones al consultarlas tienen un mejor tiempo de respuesta que en un modelado E-R con tablas normalizadas". Aquí lo que observamos es un caso típico de confusión de las capas lógica y física. El proceso de normalización, por definición, *incrementa el número de tablas lógicas.* Similarmente el E-E, que en rigor es un modelo E-R (degenerado), también es un modelo lógico. Pero el rendimiento de cualquier manejador de bases de datos no descansa en la capa lógica sino en la física. El manejador debe ser capaz de permitir las modificaciones y afinaciones que sean necesarias en la capa física sin que se vea alterada la capa lógica; en otro caso, tendremos un producto que no cumple con uno de los fundamentos principales del modelo relacional: ***la independencia lógica y física de los datos***.

Se dice también que el modelado E-R "es confuso y difícil de navegar". A esto respondemos: descubrir la esencia de los datos y presentarlos al usuario como los necesita es deber ineludible de cualquier buen modelador e implica una ardua labor de comprensión y descubrimiento donde el usuario es el actor principal. Si somos negligentes en esta labor estaremos por un lado menospreciando a los usuarios y por otro desaprovechando las oportunidades que ofrece la organización.

Sin duda los usuarios necesitan de información típica, predefinida, agregada, con cortes bien explícitos para su consulta y reacomodo a conveniencia es decir, *análisis multidimensional de datos*. Para muchos diseñadores el E-E ha sido la solución. No obstante, el modelado multidimensional de los datos puede hacerse en la capa lógica, sin comprometer la integridad y consistencia de la base de datos. Las ***vistas***, que son relaciones lógicamente definidas, sirven para este propósito al ocultar los detalles del modelo y presentando los datos conforme a cualquier contexto en particular o necesidad. Por medio

de vistas es posible representar las estructuras multidimensionales que sean requeridas y ser reaprovechadas por cualquier aplicación, herramienta profesional o como entrada a una ***verdadera base de datos multidimensional,*** tecnología desarrollada para el propósito de cubos e hipercubos.

Puesto que una vista es una representación lógica, significa que puede rápida y fácilmente modificarse sin tener que alterar los datos. Compárese lo anterior con el E-E donde es inevitable el esquema conceptual, el lógico y el físico, el software de mantenimiento y extracción, los costosos procesos de agregación y carga, amén de su administración y, por si fuera poco, el dispendioso e innecesario almacenamiento en disco. Las vistas son fundamentales en el modelo relacional, aunque los proveedores comerciales tienen diferentes implementaciones con objeto de mejorar su eficiencia: así, se habla de *vistas materializadas* (Oracle) o *vistas indizadas (SQL Server)* a fin de que, después de una primera ejecución, las siguientes tengan un óptimo tiempo de respuesta. La combinación de las *relaciones originales* y de las vistas es un explosivo ambiente capaz de potenciar a los usuarios para la creación de consultas cada vez más complejas, generando más información y conocimiento. ¡Y todo lo anterior sin comprometer ni el modelo, ni los datos almacenados, ni los recursos de almacenamiento, ni de la necesidad de software adicional, ni de onerosos procesos de actualización y mantenimiento! Cuando el E-E ofrezca algo parecido será un placer su revisión.

Para terminar este asunto, afirmamos que la construcción de un gran número de bases de datos modeladas en E-E es un mal concebido intento para dar *contexto* a las bases de datos relacionales y un cuento de nunca acabar pues, por cada necesidad de información, el fanático de este enfoque propondrá una nueva “estrella”. También, es revelador del desconocimiento que se tiene sobre el modelo relacional y sus alcances. La supuesta y aparente carencia de contexto del modelo relacional es precisamente donde descansa su inmenso poder. De hecho, a partir de una base de datos normalizada es posible extraer *cualquier contexto* de su contenido. En otras palabras, la información que se requiera, cuando se requiera y donde se requiera.

El futuro del modelo relacional

Ya desde principios de 1990 se hablaba del “fin del modelo relacional” y su sustitución por las bases de datos orientadas a objetos. Pero el caso es que en el año 2001 de 8 mil 884 millones de dólares pagados por licencias de bases de datos, 7 mil 107 correspondieron al modelo relacional. Más significativo es el hecho de que en el año 2000 las ventas para bases de datos relacionales tuvieron un incremento del 15% en tanto, para bases de datos orientadas a objetos y de otro tipo tuvieron un incremento negativo. Hay varias razones para explicar lo anterior que no detallaremos pero sin duda resaltan lo sencillo del modelo y su sólido fundamento teórico. La adición de nuevas características al modelo relacional es asunto de intenso debate así como la modernización y adecuación del lenguaje SQL a las exigencias siempre cambiantes de un entorno de gran competencia. Sin duda el modelo ha superado la prueba del tiempo. Los proveedores han añadido características de “objetos” a

sus productos, lo cual permite a los usuarios definir sus propios tipos de datos. En resumen, el modelo relacional de bases de datos es un estándar de la industria consolidado, una tecnología confiable y eficiente que estará entre nosotros aún por muchos años antes de que sea desplazada por una nueva y mejor.

Conclusión

Las necesidades de información de los usuarios cambian constantemente. No es posible anticiparlas en su totalidad. La combinación de variables junto con sus códigos en todos los proyectos de censos y encuestas de la institución conforma una cifra prácticamente infinita. El modelo relacional de bases de datos con sus relaciones normalizadas es una solución simple y elegante para satisfacer las más diversas condiciones de consulta y extracción de datos e información.

El Dr. Codd, al recibir el premio Turing en 1981, declaró que una verdadera base de datos relacional puede:

- Estar al alcance de los no programadores, donde antes los programadores fueron una necesidad.
- Incrementar la productividad de los programadores en la mayoría de aplicaciones de bases de datos.

En otro momento, hizo la siguiente reflexión: *"…es importante recordar que las bases de datos se establecen para beneficio de los usuarios finales, y no para los programadores de aplicaciones quienes actúan como intermediaros para las necesidades actuales de procesamiento de datos"*. ¡Sabias palabras!

Referencias

C. J. Date, *A tribute to Ted Codd*, may 2003

C. Imhoff, *Building the Customer-Centric Enterprise*, John Wiley & Sons; ISBN: 0471319813; 1 edition (February 1, 2001)

E. Codd, *A Relational Model of Data for Large Shared Databanks*, 1970

F. Pascal, *The Dangerous Illusion: Denormalization, Performance and Integrity*, June 2002

J. H. Orallo, *La disciplina de los sistemas de bases de datos. Historia, situación actual y perspectivas;* mayo 2002

R. Kimball, *The Data Warehouse Lifecycle Toolkit*, John Wiley, 1998

R. Armstrong, *Responding to Ralph: A rebuttal to the dimensional modeling, 1997*

R. Armstrong, *Third normal form; frequently asked question, 1997,*

W. H. Inmon, R. H. Terdenman, C. Imhof, *Exploration Warehouse*, John Wiley, 2000