

Pontificia Universidad Católica Madre y Maestra

Campus Santo Tomás de Aquino
Facultad de Ciencias de la Ingeniería



Departamento de ISC

Programa:

Base de datos I

Profesor:

Alicia Santos

Tema:

Tarea 3

Grupo:

SD-ISC-347-T-001

Integrantes:

2016-5536 - Raúl Herrera Rojas

2016-5468 - Manuel Molina

2016-5265 - Javier Falcon

Día de entrega:
12/11/2018

Inciso A

Tipo de asignación: Lectura e Investigación

Tema: Capítulo 15: Conceptos básicos de dependencias funcionales y normalización para bases de datos relacionales.

15.1 Pautas de diseño informal para esquemas relacionales.

Análisis crítico

Nos guiamos de cuatro pautas para saber si se tiene un diseño de esquema relación de calidad: asegurarse de que la semántica de los atributos está clara en el esquema, reducir la información redundante en tuplas, reducir los valores NULL en tuplas e impedir la posibilidad de generar tuplas espurias.

15.1.1 Impartir semántica clara a los atributos en las relaciones

Un esquema de relación para que sea fácil de explicar su significado. No combinar atributos de múltiples tipos de entidades y tipos de relaciones en una sola relación. Intuitivamente, si un esquema de relación corresponde a un tipo de entidad o una relación tipo, es sencillo interpretar y explicar su significado. De lo contrario, si la relación corresponde a una mezcla de múltiples entidades y relaciones, resultará confuso y la relación no se podrá explicar fácilmente.

15.1.2 Información redundante en las tuplas y actualizar anomalías

Uno de los objetivos del diseño de esquemas es minimizar el espacio de almacenamiento utilizado por las relaciones base, agrupar atributos en relación a los esquemas tienen un efecto significativo en el espacio de almacenamiento.

Si hay alguna anomalía anótalas claramente y asegúrese de que los programas que actualizan la base de datos funcionarán correctamente.

15.1.3 Valores nulos en tuplas

Si muchos de los atributos no se aplican a todas las tuplas en la relación, terminamos con muchos nulos en esas tuplas. Esto puede desperdiciar espacio en el nivel de almacenamiento y también puede llevar a problemas para entender el significado de los atributos y con operaciones concretas de JOIN en el nivel lógico. Evite colocar atributos en una relación de base cuyos valores puedan frecuentemente ser nulo, si los NULL son inevitables, hay que asegurarse

de que se aplican en solo casos excepcionales y no se aplican a la mayoría de las tuplas en la relación.

15.1.4 Generación de tuplas espurias

Se debe diseñar un esquema de relación que se pueda unir con condiciones de igualdad en atributos que están relacionados de manera adecuada, primary key y foreign key. Evite las relaciones que contienen atributos coincidentes que no son combinaciones de primary key y foreign key.

15.1.5 Resumen y discusión de las pautas de diseño

Durante el capítulo tratamos los siguientes temas: Anomalías que causan el trabajo redundante durante la inserción en y modificación de una relación, y que puede causar la pérdida accidental de información durante una eliminación de una relación, desperdicio de espacio de almacenamiento debido a NULL y la dificultad de realizar selecciones, operaciones de agregación y uniones debido a valores NULL y generación de datos no válidos y falsos durante uniones en relaciones de base con atributos coincidentes que pueden no representar una correcta primary key y foreign key.

Creatividad

Imagínate que tú y tú familia quieren construir una casa de árbol en el jardín para los niños. Tu tienes más o menos una idea de las habilidades de los integrantes de tu familia. Te toca a ti asignarle los roles a cada uno dependiendo de sus habilidades y como las interpretas. En otras palabras, la semántica.

15.2 Dependencias funcionales

Análisis critico

Se presentará una herramienta formal para el análisis de esquemas relacionales que nos permite detectar y describir algunos de los problemas mencionados anteriormente en términos precisos, siendo el concepto mas importante el de dependencias funcionales.

15.2.1 Definición de dependencia funcional

Una dependencia funcional es una restricción entre dos conjuntos de atributos de la base de datos. Definición. Una dependencia funcional, denotada por $X \rightarrow Y$, entre dos conjuntos de los atributos X e Y que son subconjuntos de R especifican una restricción en las posibles tuplas que pueden formar un estado de relación r de R . La restricción es que, para dos cualesquiera tuplas t_1 y t_2 en r que tienen $t_1[X] = t_2[X]$, también deben tener $t_1[Y] = t_2[Y]$.

Creatividad

Imaginémonos la nota final de algún estudiante en la universidad. A este estudiante se le dan un solo examen para determinar su nota final. Su nota final es totalmente dependiente de la nota

que saque en el, en otras palabras, se puede decir que la nota final del estudiante es una dependencia funcional de la nota del examen.

15.3 Formas normales basadas en claves primarias

Análisis critico

Se tratará sobre las tres primeras formas normales para los esquemas de relación y la intuición detrás de ellos. Definiciones más generales de estas formas normales, que tienen en cuenta todos los candidate keys de una relación en lugar de sólo el primary key.

15.3.1 Normalización de las relaciones

La normalización de los datos puede considerarse un proceso de análisis de la relación dada basados en sus FD y primary key para lograr las propiedades deseables de minimizando la redundancia y minimizando la inserción, eliminación y actualización de anomalías.

15.3.2 Uso práctico de las formas normales

Los proyectos de diseño más prácticos adquieren diseños existentes de bases de datos de anteriores, diseños en modelos heredados, o desde archivos existentes. Se lleva la normalización a la práctica para que los diseños resultantes sean de alta calidad y cumplan con los requisitos deseables. En la industria actual normalmente solo se llega hasta 3NF, BCNF o 4NF.

15.3.3 Definiciones de claves y atributos participando en llaves

Una key K es un super key con la propiedad adicional de que la eliminación de cualquier atributo de K hará que K no sea un super key nunca más. Si un esquema de relación tiene más de un key, cada una se denomina candidate key. Uno de los candidate keys se designan arbitrariamente para ser el primary key, y las otras son llamadas secondary keys.

15.3.4 Primera forma normal

El dominio de un atributo debe incluir solo valores atómicos y que el valor de cualquier atributo en una tupla debe ser un valor único del dominio de ese atributo. Los únicos valores de atributo permitidos por 1NF son valores atómicos o indivisibles.

15.3.5 Segunda forma normal

Un esquema de relación R está en 2NF si todos los atributos no primarios A en R dependen funcionalmente de la clave primaria de R .

15.3.6 Tercera forma normal

Análisis critico

3NF se basa en el concepto de dependencia transitiva, una dependencia funcional $X \rightarrow Y$ en un esquema de relación R es una dependencia transitiva si existe un conjunto de atributos Z en R que no es ni un candidate key ni un subconjunto de cualquier key de R y ambos $X \rightarrow Z$ y $Z \rightarrow Y$ se mantienen.

Creatividad

Imagínate que vas a alquilar un apartamento. Tienes varios candidatos que están interesados en el apartamento. Antes de elegir a uno debes evaluarlos ya que no quieres a alguien que sea mala paga. Para hacerte la vida más fácil decides tomar toda la información que tienes de cada uno y organizarla/filtrarla. En otras palabras, normalizarla.

15.4 Definición general de segunda y tercera forma normal.

Análisis crítico

El procedimiento de normalización visto hasta ahora es útil para el análisis en situaciones prácticas para una base de datos dada donde los primary keys ya han sido definidos, sin embargo, no se toma en cuenta otros candidate keys de una relación. En esta sección damos las definiciones más generales de 2NF y 3NF que tienen en cuenta todos los candidate keys de una relación.

15.4.1 Definición general de la segunda forma normal

Un esquema de relación R se encuentra en la segunda forma normal si cada atributo no primo A en R no depende parcialmente de ningún key de R. La prueba para 2NF involucra pruebas para dependencias funcionales cuyos atributos del lado izquierdo son parte de del primary key.

15.4.2 Definición general de la tercera forma normal

Un esquema de relación R se encuentra en la tercera forma normal si, siempre que la dependencia funcional no trivial $X \rightarrow A$ se mantiene en R, ya sea X es un super key de R, o A es un atributo principal de R.

15.4.3 Interpretando la definición general de la tercera forma normal

Un esquema de relación R está en 3NF si todos los atributos no primarios de R cumplen con las dos condiciones siguientes: es totalmente funcionalmente dependiente de cada key de R y es dependiente no transitivamente de cada key de R.

Creatividad

Imaginémonos que todavía no nos hemos decidido por un candidato para alquilarle el apartamento. Aun después de haber filtrado la información. Como aun no estas decidido, tomas la decisión de seguir filtrando la información a solamente lo mas importante. En otras palabras, estás haciendo una segunda o hasta una tercera normalización.

15.5 Forma normal Boyce-Codd

Análisis crítico

BCNF fue propuesto como una forma más simple de 3NF, pero este es más estricto que 3NF. Cada relación en BCNF también está en 3NF, sin embargo, una relación en 3NF no es necesariamente en BCNF.

La definición formal de BCNF difiere de la definición de 3NF en 3NF permite que A sea primo, esto no es cierto para BCNF. Eso hace que BCNF sea una forma normal más fuerte en comparación con 3NF.

Creatividad

Sigues tratando de alquilar tu apartamento. Después de haber hecho una tercera normalización todavía no estás satisfecho con la información que tienes y no has podido decidirte. Por esta razón decides utilizar la normalización de Boyce-Codd ya que esta normalización es más fuerte que la tercera.

15.6 Dependencias multivaluadas y cuarta forma normal

Análisis crítico

Si tenemos dos o más atributos multivaluados independientes en el mismo esquema de relación nos metemos en un problema de tener que repetir cada valor de uno de los atributos con cada valor del otro atributo para mantener la relación en estado consistente y mantener la independencia entre los atributos involucrados. Esta restricción está especificada por una dependencia multivaluada.

15.6.1 Definición formal de dependencia multivaluada

Un esquema de relación R está en 4NF con respecto a un conjunto de dependencias F que incluye dependencias funcionales y dependencias multivaluadas si, para toda dependencia no trivial multivaluada $X \twoheadrightarrow Y$ en F^{+17} X es un super key para R.

Creatividad

Imaginémonos la nota final de algún estudiante en la universidad. A este estudiante se le dan varios exámenes para determinar su nota final. Su nota final es totalmente dependiente de las notas que saque en los exámenes, en otras palabras, se puede decir que la nota final del estudiante es un multivalued dependency de las notas de los exámenes.

15.7 Dependencias Join y quinta forma normal

Análisis crítico

Una dependencia de join (R_1, R_2, \dots, R_n), especificada en relación esquema R , especifica una restricción en los estados r de R . La restricción establece que cada estado legal r de R debe tener una descomposición de join no aditiva en R_1, R_2, \dots, R_n . Por lo tanto, para cada uno de esos r tenemos $(\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_n}(r)) = r$.

Un esquema de relación R se encuentra en la quinta forma normal respecto a un conjunto F de funcional, multivalor y join a las dependencias si para cada dependencia de join no trivial (R_1, R_2, \dots, R_n) en F^+ , cada R_i es un super key de R .

Creatividad

Imagínate que hay una escuela con 10 cursos diferentes. Para cada curso tienes una lista de estudiantes. Pero si quisieras ver una lista de todos los estudiantes de la escuela necesitarías un joined dependency de todos los cursos.

15.8 Resumen

Análisis critico

Definimos el concepto de dependencia funcional, que es la herramienta básica para analizar esquemas relacionales y discutimos propiedades de este. Tratamos una secuencia a seguir para normalizar. Definimos de forma general la segunda y tercera forma normal dando la forma de como llevar de una a otra. Presentamos la forma normal de Boyce-Codd y vimos cómo es una forma más fuerte de 3NF. Introdujimos una cuarta forma normal basada en dependencia multivaluada. Vimos como la necesidad para descomponer a una quinta forma normal no es común y raras veces se usa en la práctica.

Creatividad

Imagínate que estas a cargo de un proyecto escolar. Para que todo salga bien debes tomar en consideración todo lo que se ha discutido previamente, como las habilidades de cada uno de los participantes al igual que organizar sus atributos para tener una mejor visualización de sus capacidades.

Inciso B

15.19 Suppose that we have the following requirements for a university database that is used to keep track of students' transcripts:

a. The university keeps track of each student's name (Sname), student number (Snum), Social Security number (Ssn), current address (Sc_addr) and phone (Sc_phone), permanent address (Sp_addr) and phone (Sp_phone), birth date (Bdate), sex (Sex), class (Class) ('freshman', 'sophomore', ... , 'graduate'), major department (Major_code), minor department (Minor_code) (if any), and degree program (Prog) ('b.a.', 'b.s.', ... , 'ph.d.'). Both Ssn and student number have unique values for each student.

b. Each department is described by a name (Dname), department code (Dcode), office number (Doffice), office phone (Dphone), and college (Dcollege). Both name and code have unique values for each department.

c. Each course has a course name (Cname), description (Cdesc), course number (Cnum), number of semester hours (Credit), level (Level), and offering department (Cdept). The course number is unique for each course.

d. Each section has an instructor (Iname), semester (Semester), year (Year), course (Sec_course), and section number (Sec_num). The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the total number of sections taught during each semester.

e. A grade record refers to a student (Ssn), a particular section, and a grade (Grade).

Design a relational database schema for this database application. First show all the functional dependencies that should hold among the attributes. Then design relation schemas for the database that are each in 3NF or BCNF. Specify the key attributes of each relation. Note any unspecified requirements and make appropriate assumptions to render the specification complete.

- **Esquemas iniciales (0NF).**

Student

Sname	<u>Snum</u>	<u>Ssn</u>	Sc_addr	Sc_phone	Sp_addr	Sp_phone	Bdate	Sex	Class	Major_code	Minor_code	Prog
-------	-------------	------------	---------	----------	---------	----------	-------	-----	-------	------------	------------	------

Department

Dname	<u>Dcode</u>	Doffice	Dphone	Dcollege
-------	--------------	---------	--------	----------

Course

Cname	Cdesc	<u>Cnum</u>	Credit	Level	Cdept
-------	-------	-------------	--------	-------	-------

Section

Iname	Semester	Year	<u>Cnum</u>	Sec_num	<u>Id_sec</u>
-------	----------	------	-------------	---------	---------------

Grade

<u>Ssn</u>	Grade	<u>Id_sec</u>
------------	-------	---------------

Instructor

Iname	<u>Inum</u>	<u>Ssn</u>	Ic_addr	Ic_phone	Ip_addr	Ip_phone	Bdate	Sex
-------	-------------	------------	---------	----------	---------	----------	-------	-----

Dependencias funcionales:

- Student
 - FD1: {Ssn} -> {Sname, Sp_addr, Sp_phone, Bdate, Sex}
 - FD2: {Snum} -> {Sc_phone, Sc_addr, Class, Major, Minor, Prog}
- Department
 - FD3: {Dcode} -> {Dname, Doffice, Dphone, Dcollege}
- Curso
 - FD4: {Cnum} -> {Cname, Cdesc, Credit, Level, Cdept}
- Section
 - FD5: {SSN, Id_sec} -> {Iname}
- Instructor

- FD6: {Ssn} -> {Iname, Ip_add, Ip_phone, Sex, Bdate}
- FD7: {Inum} -> {Ic_phone, Ic_addr}

Normalización

- Student: La relación Student es llevada a 1NF debido a sus atributos compuestos de Sname, Sc_addr y Sp_addr. Luego, es llevada a la 2NF para separar los atributos que dependen de las dos llaves diferentes. Finalmente, la relación ya queda en 3NF debido a que no se forma ninguna dependencia transitiva.

1NF

<u>Snum</u>	SSN	Sf_name	Sl_name	Sc_st	Sc_house	Sc_zip	Sc_phone	Sp_st	Sp_house	Sp_zip	Sp_phone	Bdate	Sex	Major	Minor	Prog
-------------	-----	---------	---------	-------	----------	--------	----------	-------	----------	--------	----------	-------	-----	-------	-------	------

2NF y 3NF

<u>Snum</u>	Sc_st	Sc_house	Sc_zip	Sc_phone	Major	Minor	Class	Prog	...SSN...
-------------	-------	----------	--------	----------	-------	-------	-------	------	-----------

<u>SSN</u>	Sf_name	Sl_name	Sp_st	Sp_house	Sp_zip	Sp_phone	Bdate	Sex
------------	---------	---------	-------	----------	--------	----------	-------	-----

- Instructor: En un caso muy similar a la relación Student, encontramos que Instructor debe sufrir transformaciones a 1NF y a 2NF. La forma normal 3 se consigue automáticamente después de transformar a la forma normal 2 debido a la inexistencia de dependencias transitivas.

1NF

<u>Inum</u>	<u>SSN</u>	If_name	IL_name	Ic_st	Ic_house	Ic_zip	Ic_phone	Ip_st	Ip_house	Ip_zip	Ip_phone	Bdate	Sex
-------------	------------	---------	---------	-------	----------	--------	----------	-------	----------	--------	----------	-------	-----

2NF y 3NF

<u>Inum</u>	Ic_st	Ic_house	Ic_zip	Ic_phone	..SSN..
-------------	-------	----------	--------	----------	---------

<u>SSN</u>	If_name	IL_name	Ip_st	Ip_house	Ip_zip	Ip_phone	Bdate	Sex
------------	---------	---------	-------	----------	--------	----------	-------	-----

- Section: finalmente, la relación Section debe ser llevada a 1NF por el atributo compuesto Iname. Luego de esto, también queda en estado 2NF ya que no existen dependencias parciales y por eso no se toma en cuenta. Para terminar, fue llevada a la forma 3NF por la dependencia transitiva Inum -> Iname. En este caso, la relación resultante de la transformación es obviada porque ya existe una relación Instructor, por lo que solamente se señala el Foreign key de Inum.

1NF y 2NF

<u>Id_sec</u>	Sec_num	Cnum	Semester	Year	Inum	If_name	IL_name
---------------	---------	------	----------	------	------	---------	---------

3NF

<u>Id_sec</u>	Sec_num	Cnum	Semester	Year	Inum..
---------------	---------	------	----------	------	--------

Las relaciones Department, Course y Grade ya se encuentran en la forma normal 3.

- **Asunciones:**
 - Se agregó una tabla instructor debido a que consideramos que es una entidad que debe estar presente, ya que se hace referencia al instructor en la relación Section.
 - En la relación Section se agregó un atributo Id_sec para tener un valor único en cada sección y evitar tener que diferenciarlas a través del semestre y el año.
 - El atributo Sec_course fue cambiado a Cnum ya que queda una relación más clara entre sección y curso que antes podía confundirse.

Bibliografía

Elmasri Navathe. (2010). FUNDAMENTOS DE BASES DE DATOS. 6th ed. London: Pearson