TEMA 2: El modelo relacional.

En este tema expondremos sucintamente el concepto general de modelo de datos, incidiendo en los conceptos de estática y dinámica de un modelo de datos. Posteriormente analizaremos de forma detallada el modelo de datos que emplean las bases de datos relacionales, esto es, el modelo relacional.

Analizaremos con profundidad el modelo relacional, exponiendo su estructura, el concepto de valor nulo en el modelo relacional y la lógica trivaluada y cuatrivaluada del modelo relacional, concluyendo con la exposición del álgebra relacional y los operadores que formalizan el modelo de consultas en una base de datos relacional.

2.1 Concepto de modelo de datos.

Podemos decir, como primera aproximación, que un *modelo de datos* es un conjunto de conceptos que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos, a la cual denominamos *esquema*. Según el nivel de abstracción en el que se encuentre la estructura descrita, el modelo que permite su descripción será un *modelo externo* (visión que tiene de la base de datos cada usuario en particular), *modelo global* (enfoque del conjunto de la empresa) o *modelo interno* (forma en que se organizan los datos en el almacenamiento físico). Los modelos externos nos permiten representar los datos que necesita cada usuario en particular con las estructuras propias del lenguaje de programación que va a emplear. Los modelos globales ayudan a describir los datos para el conjunto de usuarios, podríamos decir que es la información a nivel de empresa; y, por último, los modelos internos están orientados a la máquina, siendo sus elementos de descripción punteros, índices, agrupamientos, etc.

Los modelos que tienen un mayor interés son los modelos globales, ya que los externos utilizan los mismos conceptos y los internos no están estandarizados ni existen en realidad como tales modelos, sino que son propios de cada uno de los productos comerciales. Los modelos globales se clasifican, a su vez, en conceptuales y convencionales.

- Los *modelos conceptuales* (también denominados de alto nivel) facilitan la descripción global del conjunto de información de la empresa con independencia de la máquina, por lo que sus conceptos son cercanos al mundo real (entidades, atributos, etc.); son modelos de análisis, no de implementación.

- Los *modelos convencionales* se encuentran soportados por los SGBD y están orientados a describir los datos a nivel lógico para el SGBD (de ahí que también reciban el nombre de modelos de bases de datos), por lo que sus conceptos son propios de cada SGBD (tablas o relaciones en el modelo relacional, redes en el modelo en red, árboles en el jerárquico, etc.).

El *modelo de datos* es el instrumento que se aplica para obtener el *esquema*, entendiendo como esquema la descripción de la estructura de la base de datos. Es preciso distinguir entre esquema y *ocurrencia del esquema*, que son los datos que se encuentran almacenados en un esquema en un momento determinado. El esquema no varía mientras no varíe el mundo real que éste describe; en tanto que una ocurrencia del esquema, es decir, los datos contenidos en él, son distintos en el transcurso del tiempo.

Una vez llegados a este punto, podemos definir de forma más precisa, un modelo de datos como un *conjunto de conceptos, reglas y convenciones que nos permiten describir y manipular (consultar y actualizar) los datos de un cierto mundo real que deseamos almacenar en la base de datos.*

Las propiedades del modelo de datos son de dos tipos: *estáticas*, o relativamente invariantes con el tiempo, que responden a lo que se suele entender como estructura; y *dinámicas*, que son las operaciones que se aplican a los datos o valores almacenados en las estructuras, los cuales varían en el transcurso del tiempo al aplicárseles dichas operaciones.

2.1.1 Estática de un modelo de datos.

La estática de un modelo de datos está compuesta por elementos permitidos y elementos no permitidos o restricciones. Los elementos permitidos difieren según el modelo de datos, pero en general son:

- Objetos (entidades, relaciones, registros, etc.).
- Asociaciones entre objetos (interrelaciones, conjuntos, etc.).
- Propiedades o características de los objetos o de las asociaciones (atributos, campos, elementos de datos, etc.).
- Dominios, conjuntos nominados de valores sobre los que se definen las propiedades.

La representación de estos elementos depende de cada modelo de datos, pudiendo hacerse en forma de grafos (como en el modelo jerárquico o el modelo en red) o de tablas (como en el modelo relacional), o bien en ambos (como en el modelo entidad/interrelación).

Los elementos no permitidos o restricciones son debidos a que no todos los valores, cambios de valor o estructuras están permitidos en el mundo real. Por ejemplo, un niño de dos años no puede estar viudo, etc. Además, cada modelo de datos también impone por sí mismo limitaciones a las estructuras que admite; así, el modelo relacional no permite que dos filas de la misma tabla sean iguales.

Estas limitaciones, que unas veces las impone el modelo de datos y otras vienen impuestas por el universo del discurso que estamos modelando se denominan *restricciones*. Las restricciones impuestas por el modelo son *restricciones inherentes*, y las que responden a que el sistema de información es un reflejo del mundo real son *restricciones de integridad o semánticas*. Las restricciones inherentes son propias del modelo y, por tanto, varían de un modelo a otro. Por el contrario, las restricciones de integridad son facilidades que se ofrecen al diseñador a fin de que pueda representar en el esquema, lo más fielmente posible, la semántica de los datos.

2.1.2 Dinámica de un modelo de datos.

Los valores que toman los distintos objetos de un esquema en un momento determinado t_i reciben el nombre de ocurrencia del esquema o estado de la base de datos en el tiempo t_i (BD_i); en otro momento t_j la ocurrencia del esquema será (BD_j). Si entre t_i y t_j se ha producido un cambio en algún valor de la base de datos (alta, baja o modificación) $BD_i \neq BD_j$. Tanto BD_i como BD_j deben ser ocurrencias válidas de la base de datos, es decir, ambas deben cumplir las restricciones de integridad así como las posibles restricciones asociadas al cambio de estado.

La componente dinámica del modelo consta de un conjunto de operadores que se definen sobre la estructura del correspondiente modelo de datos, ya que no todas las estructuras admiten el mismo tipo de operaciones. La aplicación de una operación a una ocurrencia de un esquema transforma a ésta en otra ocurrencia:

$$O(BD_i) = BD_j.$$

Pudiendo ser $BD_i = BD_j$, por ejemplo, en caso de consulta o cuando falla la operación por haberse producido un error. Una operación tiene dos componentes:

1. *Localización o enfoque*, consistente en localizar una ocurrencia de un objeto indicando un camino, o bien un conjunto de ocurrencias especificando una condición.

2. *Acción*, que se realiza sobre la(s) ocurrencia(s) previamente localizada(s) mediante una operación de localización, y puede consistir en una recuperación o en una actualización (inserción, borrado o modificación).

La distinción entre localización y acción es de tipo formal, existiendo esta diferenciación en algunos lenguajes como el LMD de Codasyl, o estando reunidas ambas operaciones en otros como el SQL.

2.2 El modelo relacional.

La aparición del artículo de Codd sobre los SGBD relacionales supuso en la práctica la proposición de un nuevo modelo de datos basado en la teoría de las relaciones, donde los datos se estructuran lógicamente en forma de relaciones (tablas), siendo el objetivo fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a cualquier otra característica de tipo físico. El modelo propuesto por Codd presentaba los siguientes objetivos:

- *Independencia física*. El modo en que se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a los datos no han de modificar sus programas por cambios en el almacenamiento físico.

- *Independencia lógica*. Añadir, eliminar o modificar cualquier elemento de la base de datos no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos.

- *Flexibilidad*. Poder ofrecer a cada usuario los datos de la forma más adecuada a su aplicación.

- *Uniformidad*. Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.

- *Sencillez*. Las características anteriores, así como unos lenguajes de usuario sencillos, producen el resultado de que el modelo de datos relacional es fácil de comprender y de utilizar.

2.2.1 Estructura del modelo relacional.

El modelo relacional está basado en la relación como elemento básico y puede ser representado como una tabla (figura 2.2.1.1).

NOMBRE

atributo 1	atributo 2	…	atributo n	
XXX	XXX	…	XXX	tupla 1
XXX	XXX	…	XXX	tupla 2
…	…	…	…	…
XXX	XXX	…	XXX	tupla m

Figura 2.2.1.1: Representación de una relación en forma de tabla.

En la figura podemos distinguir su *nombre*, un conjunto de columnas, denominadas *atributos*, que representan propiedades de la tabla y que también están caracterizadas por su nombre, y un conjunto de filas llamadas *tuplas*, que contienen los valores que toma cada uno de los atributos para cada elemento de la relación.

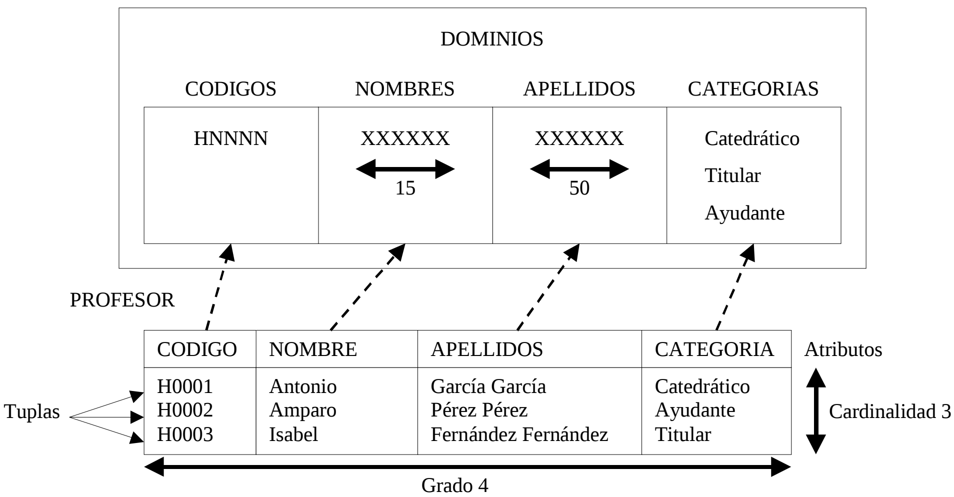

Figura 2.2.1.2: Representación de la relación profesor.

La figura 2.2.1.2 representa la relación PROFESOR, donde aparece la estructura del modelo relacional. En ella se puede observar el nombre de la relación (PROFESOR); los atributos (código, nombre, apellidos, categoría); los dominios (de donde los atributos toman sus valores, permitiéndose que varios atributos pueden tomar valores del mismo dominio); las tuplas (cada una de las cuales contiene los valores que toma el código, nombre, apellidos, etc.); el grado (número de atributos) y la cardinalidad (número de tuplas).

Conviene aclarar en este punto el concepto de dominio expuesto en el párrafo anterior. Un dominio es un conjunto finito de valores homogéneos y atómicos caracterizado por un nombre. Los valores son homogéneos porque son todos del mismo tipo y atómicos porque son indivisibles en el modelo, pues si se descomponen pierden la semántica asociada a los mismos. Así, si hablamos del dominio de nacionalidades, la nacionalidad "española" es atómica, pues si se descompone en sus letras pierde el sentido semántico, ya que las letras aisladas pierden el significado de una nacionalidad.

Además, conviene resaltar que aunque una relación se represente en forma de tabla, tiene una serie de elementos que la distinguen de una tabla, ya que no se admiten filas duplicadas, las filas y columnas no están ordenadas y es plana, es decir, en el cruce de una fila y de una columna sólo puede haber un valor.

La representación en forma de tabla ha conducido a que los usuarios utilicen el nombre de *tabla* para denominar las relaciones y, como consecuencia de ello, se llame *filas* a las tuplas y *columnas* a los atributos. En la figura 2.2.1.3 se tiene una comparación de la terminología de relación, tabla y fichero.

RELACIÓN	TABLA	FICHERO
TUPLA	FILA	REGISTRO
ATRIBUTO	COLUMNA	CAMPO
GRADO	Nº DE COLUMNAS	Nº DE CAMPOS
CARDINALIDAD	Nº DE FILAS	Nº DE REGISTROS

Figura 2.2.1.3: Comparación de la terminología de relación, tabla y fichero.

2.2.2 Concepto de valor nulo en el modelo relacional.

Si bien los valores nulos no son un concepto exclusivo del modelo relacional, ha sido en el contexto de este modelo donde se ha abordado su estudio de manera más sistemática y donde se están realizando más investigaciones a fin de formalizar su tratamiento.

Se puede definir el valor nulo (también denominado valor ausente) como una señal utilizada para representar información desconocida, inaplicable, inexistente, no válida, no proporcionada, indefinida, etc. Su necesidad en las bases de datos es evidente por diversas razones:

- Crear tuplas (filas) con ciertos atributos desconocidos en ese momento, por ejemplo, el año de edición de un libro.

- Añadir un nuevo atributo a una relación existente; atributo que, en el momento de añadirse, no tendría ningún valor para las tuplas de la relación.

- Atributos inaplicables a ciertas tuplas, por ejemplo, la editorial para un artículo o la profesión para un menor.

El tratamiento de valores nulos exige definir operaciones de comparación, operaciones aritméticas, operaciones algebraicas y, funciones de agregación, de forma específica para el caso de que alguno de los operandos tome valores nulos, y obliga también a introducir nuevos operadores especiales. En las operaciones de comparación se plantea el problema de saber si dos valores nulos son o no iguales. No podemos decir que es cierto que sean iguales puesto que estaríamos afirmando que no son "tan" desconocidos; pero tampoco podemos decir que es falso que sean iguales; la única solución que nos queda es decir que *quizás* sean iguales. Surge, de esta manera, una lógica distinta de la habitual (L2V), la denominada lógica trivaluada (L3V).

2.2.2.1 Lógica trivaluada.

Como hemos visto la lógica trivaluada surge de la necesidad de representanción de los valores nulos en el modelo relacional. Por ello surgen nuevas tablas de verdad y tres nuevos operadores que actúan sobre los valores nulos.

En la figura 2.2.2.1.1 aparecen las tablas de verdad para la lógica trivaluada, donde existen los valores C (cierto), F (falso) y Q (quizás).

AND	C	Q	F
C	C	Q	F
Q	Q	Q	F
F	F	F	F

OR	C	Q	F
C	C	C	C
Q	C	Q	Q
F	C	Q	F

NOT	
C	F
Q	Q
F	C

Figura 2.2.2.1.1: Tablas de verdad de la lógica trivaluada.

Además, los tres nuevos operadores que actúan sobre los valores nulos en la lógica trivaluada son:

- El operador "ES_POSIBLE" ("MAYBE"), que aplicado al valor de una expresión "quizás" da como resultado "cierto".

- El operador "ES_NULO" ("IS_NULL"), que toma el valor cierto si el operando es nulo y falso en caso contrario.

- El operador "SI_NULO" ("IF_NULL"), que se aplica a dos operandos y devuelve el valor del primero, salvo que sea nulo, en cuyo caso devuelve el valor del segundo.

En cuanto a las operaciones aritméticas con valores nulos, se considera nulo el resultado de sumar, restar, multiplicar o dividir cuando alguno de los operandos toma valor nulo.

La lógica trivaluada puede causar problemas a los usuarios pues atenta contra la intuición. En efecto, si un usuario realiza las consultas siguientes: Libros editados en 1990, libros editados antes de 1990 y libros editados después de 1990, tendría, en nuestra lógica habitual, que haber recuperado todos los libros de la base de datos; pero esto puede no ser así, ya que no habría recuperado los libros cuyo año de edición se desconoce, es decir, cuyo año de edición es nulo.

Por ello, expresiones como X = X, siendo X una variable o p OR NOT(p), siendo p una expresión condicional, que son siempre ciertas en el mundo real, no se cumplen en la L3V (pues X puede ser quizás y la tautología p OR NOT(p) no cubre el caso quizás), lo que afecta a las equivalencias algebraicas que se utilizan en las leyes de transformación en la optimización de consultas.

El propio Codd afirma que esta parte del modelo relacional no presenta un fundamento teórico tan sólido como las otras, por lo que propone añadir a L3V capacidades de inferencia que permitan detectar tautologías, como por ejemplo p OR NOT(p) OR quizás.

Los problemas de la L3V se agravan aún más si cabe en la práctica, ya que los productos relacionales no siempre implementan los valores nulos de una manera consistente. Por estos motivos algunos expertos propugnan evitar los valores nulos mediante la utilización de *valores por defecto*. Este enfoque presenta los siguientes problemas:

- No resuelve toda la problemática asociada al tratamiento de valores nulos, sino que simplemente proporciona un medio para representar información desconocida.

- Al representarse mediante un valor (y no por medio de un concepto especial), fuerza a comprobar las dependencias funcionales y otro tipo de dependencias al introducir la información.

- La representación del hecho de que un valor es desconocido, no sólo depende del tipo de datos de cada columna, sino que también puede variar entre columnas del mismo tipo.

- Las distintas técnicas para tratar estos valores por defecto quedan embebidas en los programas, con pocas posibilidades de que sean uniformes y sistemáticas y estén bien documentadas.

- Cada valor desconocido se trata como si fuera cualquier otro valor.

- Representa un retroceso hacia enfoques no sistemáticos y ad hoc propios de la era pre-relacional.

Además, la L3V no diferencia entre los valores nulos que corresponden a que no es posible aplicar un atributo de los valores nulos que corresponden a que no es conocido el valor del atributo actualmente, pero dicho atributo si es aplicable. Por ello surge el concepto de lógica cuatrivaluada.

2.2.2.2 Lógica cuatrivaluada.

La lógica cuatrivaluada surge de la necesidad de diferenciar dos tipos importantes de valores nulos: *inaplicables*, esto es, que no tienen sentido, de *desconocidos aplicables*, es decir, que momentáneamente son desconocidos pero deberían existir y puede que, en un determinado momento, lleguen a conocerse. Esta diferencia, que puede parecer académica en exceso, es de hecho muy importante para reflejar con más precisión la semántica del universo del discurso a tratar y para responder de manera más inteligente a las consultas que se realicen sobre la base de datos.

En la lógica cuatrivaluada se distingue, por tanto, entre un valor nulo o marca que representa información desconocida pero aplicable, que denominaremos “a”, de otro que representa información inaplicable, que denominaremos “i”. En este caso las operaciones aritméticas quedan modificadas como sigue, siendo & un operador aritmético (suma, resta, multiplicación o división) y “x” un valor no nulo de la base de datos:

x & a = a & x = a
x & i = i & x = i
a & i = i & a = i
a & a = a
i & i = i

En la figura siguiente (figura 2.2.2.2.1) se muestran las tablas de verdad para la lógica cuatrivaluada, con los valores C (cierto), A (aplicable), F (falso) e I (inaplicable).

AND	C	A	F	I
C	C	A	F	I
A	A	A	F	I
F	F	F	F	F
I	I	I	F	I

OR	C	A	F	I
C	C	C	C	C
A	C	A	A	A
F	C	A	F	F
I	C	A	F	I

NOT	
C	F
A	A
F	C
I	I

Figura 2.2.2.2.1: Tablas de verdad de la lógica cuatrivaluada.

Sin embargo, la lógica soportada por los SGBD es la lógica trivaluada, quedando la lógica cuatrivaluada como un formalismo.

2.3 Álgebra relacional.

La dinámica del modelo relacional, en lo que al álgebra se refiere, constituye una colección de operadores de alto nivel que, aplicados a las relaciones, dan como resultado nuevas relaciones. Sean R y R’ dos relaciones y O un operador cualquiera del álgebra relacional. Una operación del álgebra relacional consiste en aplicar O a la relación R, obteniéndose R’. Al ser el resultado de la operación otra relación, se cumple la *propiedad de cierre*, es decir, si $O_1 \ldots O_n$ representan operadores, se cumple:

$$O_n(\ldots(O_1(R)) = R'$$

Los operadores del álgebra son siempre relacionales y los operandos se aplican por tanto, a relaciones a fin de formular consultas a la base de datos. Originalmente, Codd definió, cuando propuso el modelo relacional, ocho operadores para el álgebra relacional, divididos en dos grupos de cuatro:

- Los operadores tradicionales de conjuntos: *unión, intersección, diferencia y producto cartesiano*.

- Los operadores relacionales especiales: *restricción, proyección, combinación y división*.

Aparte de esta clasificación, los operadores se pueden dividir también en operadores primitivos y operadores derivados:

- Los operadores primitivos son los operadores esenciales que no pueden obtenerse de otros (sin ellos, el álgebra relacional no sería un lenguaje completo).

- Los operadores derivados se pueden obtener aplicando varios de los operadores primitivos. Podríamos prescindir de ellos, ya que no aportan nada nuevo; sin embargo, son muy útiles y simplifican muchas operaciones habituales.

Otra posible clasificación, atendiendo al número de operandos de cada operador, los divide en:

- Unarios, si el operador tiene una única relación como operando.

- Binarios, si el operador tiene dos relaciones como operandos.

Es posible definir, y así se ha hecho con posterioridad, más operadores aparte de los ocho originales, siempre que tengan una o dos relaciones como operandos y una relación como resultado.

Veremos a continuación los principales operadores del álgebra relacional, dando su definición formal y un ejemplo que clarifique su uso.

2.3.1 Operación de asignación y renombrado de atributos.

Antes de empezar a definir los operadores del álgebra relacional, vamos a introducir una operación auxiliar: la *asignación*, mediante la cual también se puede llevar a cabo el renombrado de atributos. La operación asignación se utiliza para almacenar el resultado de una consulta en una nueva relación o para denominar resultados intermedios cuando se desea dividir una única operación compleja en una secuencia de operaciones más simples; también se emplea para asignar un nuevo nombre a una relación existente (cambiando, por ejemplo, los nombres de sus atributos). El símbolo con el que se suele representar la operación de asignación es una flecha que señala hacia la nueva relación a la que se asignará el resultado de la operación:

$$\text{RELACION\_NUEVA} \leftarrow \text{O(R)}$$

O(R) es el resultado de una o más operaciones algebraicas; aunque también podría aplicarse este operador para almacenar una relación en otra nueva relación con nombre distinto, es decir:

$$R' \leftarrow R$$

La operación de renombrado consiste en asignar nuevos nombres a los atributos de una relación (la cual puede ser el resultado de una operación algebraica); el renombrado es una operación necesaria, como veremos más adelante, en ciertos casos en los que intervienen operadores binarios, y a veces es preciso cambiar previamente en todo, o en parte, los nombres de los atributos de alguna de las relaciones que actúan como operandos. La forma de llevar a cabo el renombrado de los atributos es realizando una operación de asignación en la cual se especifican los nombres de los atributos de la relación que se encuentra a la izquierda del símbolo de asignación:

$$\text{RELACION\_NUEVA}(A_1,A_2,\ldots,A_n) \leftarrow O(R)$$

También en este caso, en lugar de O(R) se puede escribir el nombre de una relación.

2.3.2 Operador primitivo unario restricción (σ).

Antes de comentar este operador, es necesario señalar que los operadores unarios tienen como operando una única relación; para su definición, utilizaremos la siguiente notación:

Sea R(A) un esquema de relación cuyo contexto es el conjunto de atributos A definidos sobre el conjunto de dominios D:

$$R(A) = R(A_1 : D_1, A_2 : D_2, \ldots A_n : D_n)$$

La relación r(R), definida sobre el esquema R, de grado n y cardinalidad m estará constituida por un conjunto de m tuplas:

$$r(R) = \{t_i\}_{i=}^{m} \quad \text{donde } t_i = < v_{i1}, v_{i2}, \ldots, v_{in} > / \; v_{ij} \in D_j$$

La restricción, también llamada selección, de una relación mediante una expresión lógica (predicado de selección), da como resultado una relación formada por el subconjunto de tuplas que satisface dicha expresión. La relación resultante constituye un subconjunto horizontal de r(R).

Formalmente, sea θ un operador de comparación ($<, >, =, \Leftrightarrow, \geq, \leq, \neq$) y p un predicado de selección, formado por una expresión lógica integrada por cláusulas de la forma: $A_i \ \theta \ A_j$ o $A_i \ \theta$ constante, unidas por los operadores booleanos AND, OR, NOT. El operador selección (aplicado a la relación R con el predicado p) se denota $\sigma_p(R)$ y produce una relación, cuyo esquema será el mismo (es decir, R) y cuya extensión r' será:

$$r'(R) = \{ t_i \in r(R) / p(t_i) = \text{“cierto”}\}$$

El grado de la relación resultante será por tanto n, es decir, el mismo que el de la relación R y su cardinalidad m’ $\Leftrightarrow$ m.

La aplicación consecutiva del operador de selección a una relación, $\sigma_{pn}(\sigma_{pn-1}\ldots(\sigma_1(R)))$,es igual a una única operación de restricción con todos los predicados p_1, p_2,…, p_n, unidos por el operador booleano AND: $\sigma_{p1 \text{ y } p2 \ldots \text{ y } pn}$.

En la figura 2.3.2.1 puede verse un ejemplo de aplicación del operador restricción.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

$\sigma_{categoria=ayudante}$(PROFESOR)

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0002	Amparo	Pérez Pérez	Ayudante
H0004	José	Hernández Hernández	Ayudante

Figura 2.3.2.1: Ejemplo de aplicación del operador restricción.

2.3.3 Operador primitivo unario proyección (π).

La proyección de una relación sobre un subconjunto de sus atributos es una relación definida sobre ellos, eliminando las tuplas duplicadas que hubieran podido resultar; es, por tanto, un subconjunto vertical de la relación a la que se aplica el operador.

Formalmente, sea X un subconjunto estricto y no vacío de A: $X \subset A$ y $X \neq \phi$, la aplicación del operador de proyección π a R en el contexto de X, que se denota por $\pi_x(R)$ será una relación r’ cuyo esquema es R’(X) y cuya extensión es el conjunto de tuplas de la relación original definidas sobre los atributos X, eliminando las que resultan duplicadas, es decir:

$$r'(R') = \{t_i(X) \;/\; X \subset A\}$$

El grado n’ y la cardinalidad m’ de la relación resultante cumplen n’<n y m’≤ m, siendo n y m el grado y la cardialidad de la relación original, respectivamente.

En la figura 2.3.3.1 puede verse un ejemplo de aplicación del operador proyección.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

$$\pi_{categoria}(\mathrm{PROFESOR})$$

CATEGORIA
Catedrático
Ayudante
Titular

Figura 2.3.3.1: Ejemplo de aplicación del operador proyección.

2.3.4 El operador primitivo binario unión (∪).

Antes de comentar este operador, es necesario señalar que los operadores binarios se aplican a dos relaciones, y algunos de ellos (unión, diferencia e intersección) exigen que las dos relaciones involucradas sean *compatibles* en sus esquemas. Se dice que dos esquemas de relación R1 ($A1_i : D1_i$) y R2 ($A2_i : D2_i$), con cardinalidades m_1 y m_2, son compatibles a efectos de dichos operadores, cuando ambos están definidos sobre el mismo conjunto de dominios, cumpliéndose:

$$\forall\ A1_i \ \exists\ A2_j \ /\ dom(A1_i) = dom(A2_j) \text{ y } \forall\ A2_j \ \exists\ A1_i \ /\ dom(A2_j) = dom(A1_i)$$

Es decir, R1 y R2 serán semánticamente equivalentes, lo que no quiere decir que los nombres de los atributos sean los mismos, sino que han de estar definidos sobre los mismos dominios.

Si los nombres de los atributos de las dos relaciones son distintos o están en distinto orden, es decir, no hay correspondencia uno a uno entre los nombres de los atributos de los esquemas de R1 y R2, será precisa una operación de renombrado de los atributos del esquema de la relación resultante.

Por ejemplo, en la figura 2.3.4.1 tenemos dos relaciones que son compatibles en sus esquemas al estar sus atributos definidos sobre los mismos dominios; sin embargo, los nombres de los atributos no coinciden. Se puede proceder a la operación de renombrado de PDI1:

$$\mathrm{PDI}(codigo, nombre, apellidos, categoria) \leftarrow \mathrm{PDI1}$$

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular

PDI1

CODIGO	NOMBRE	APELLIDOS	TRABAJO
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

Figura 2.3.4.1: Ejemplo de dos relaciones compatibles en sus esquemas en las que los nombres de los atributos no coinciden.

La unión de dos relaciones r_1 y r_2 con esquemas compatibles R1 y R2 es otra relación definida sobre el mismo esquema de relación y cuya extensión estará constituida por el conjunto de tuplas que pertenezcan a r_1 o a r_2 (se eliminarán las tuplas duplicadas puesto que se trata de un conjunto).

Formalmente, sean dos relaciones r_1 y r_2 con esquemas R1 y R2 compatibles, la unión de ambas, denotada por: R1∪R2 será una relación con esquema R (R es igual a R1 o a R2 ya que ambos son equivalentes) y con extensión r:

$$r(R) = \{t_i \,/\, t_i \in r_1 \vee t_i \in r_2\}$$

En la figura 2.3.4.2 puede verse un ejemplo de aplicación de la operación unión.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular

PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

PROFESOR $\cup$ PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

Figura 2.3.4.2: Ejemplo de aplicación de la operación unión.

2.3.5 El operador primitivo binario diferencia (-).

La diferencia de dos relaciones r_1 y r_2 con esquemas compatibles R1 y R2 es otra relación definida sobre el mismo esquema de relación y cuya extensión estará constituida por el conjunto de tuplas que pertenezcan a r_1 pero no a r_2.

Formalmente, sean dos relaciones compatibles, con esquemas R1 y R2, la diferencia de ambas denotada por R1-R2 será una relación con esquema R (R es igual a R1 o a R2 ya que ambos son equivalentes) y con extensión r:

$$r(R) = \{t_i \,/\, t_i \in r_1 \wedge t_i \notin r_2\}$$

En la figura 2.3.5.1 puede verse un ejemplo de aplicación de la operación diferencia.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular

PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

PROFESOR - PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante

Figura 2.3.5.1: Ejemplo de aplicación de la operación diferencia.

2.3.6 El operador primitivo binario producto cartesiano generalizado (x).

El producto cartesiano generalizado de dos relaciones de cardinalidades m_1 y m_2 es una relación definida sobre la unión de los atributos de ambas relaciones y cuya extensión estará constituida por las m_1 x m_2 tuplas formadas concatenando cada tupla de la primera relación con cada una de las tuplas de la segunda (obsérvese que aquí no se exige que las dos relaciones sean compatibles en sus esquemas).

Formalmente, sean las relaciones r_1 y r_2 con esquemas R1 y R2 el producto de ambas , denotado R1 x R2, será una relación de grado n_1+n_2 cuyo esquema R estará formado por los n_1+n_2 atributos $A1 \cup A2$, es decir:

$$(A1_i : D1_i ,\ldots, A1_{n1} : D1_{n1}, A2_{i'} : D2_{i'} ,\ldots, A2_{n2} : D2_{n2})$$

Y cuya extensión, de cardinalidad m x m' sera:

$$\{<V1_{i1},\ldots,V1_{in1},V2_{j1},\ldots,V2_{jn2}> / \forall i \ \forall j \ (<V1_{i1},\ldots,V1_{in1}> \in r_1 \wedge < V2_{j1},\ldots,V2_{jn2}> \in r_2)\}$$

En la figura 2.3.6.1 puede verse un ejemplo de aplicación del operador producto cartesiano generalizado.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante

ASIGNATURA

NUMERO	DESCRIPCION
10000	Tratamiento de datos
10001	Análisis estadístico
10002	Cálculo numérico

PROFESOR X ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	NOMBRE
H0001	Antonio	García García	Catedrático	10000	Tratamiento de datos
H0001	Antonio	García García	Catedrático	10001	Análisis estadístico
H0001	Antonio	García García	Catedrático	10002	Cálculo numérico
H0002	Amparo	Pérez Pérez	Ayudante	10000	Tratamiento de datos
H0002	Amparo	Pérez Pérez	Ayudante	10001	Análisis estadístico
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico

Figura 2.3.6.1: Ejemplo de aplicación de la operación producto cartesiano generalizado.

2.3.7 El operador derivado binario combinación (|X|).

La combinación (denotada por |X|) de dos relaciones respecto a una cierta condición de combinación, es otra relación constituida por todos los pares de tuplas t_i y t_j concatenadas, tales que, en cada par, las correspondientes tuplas satisfacen la condición especificada. La forma general de expresar la combinación de dos relaciones

$R1(A1_1, A1_2,\ldots, A1_{n1})$ y $R2(A2_1, A2_2,\ldots, A2_{n2})$ es: $\underset{\text{(condición)}}{R_1|X|R_2}$

La condición de combinación, en el caso más sencillo, está referida a dos atributos $A1_i$ y $A2_j$, cada uno de los cuales pertenece a una de las dos relaciones unidos por un operador de comparación θ: $A1_i\ \theta\ A2_j$.

En el caso más general, un par de atributos puede ir unido a otro par de atributos mediante el operador booleano AND:

$$A1_{i1}\ \theta\ A2_{j1} \text{ AND } A1_{i2}\ \theta\ A2_{j2} \text{ AND } \ldots \text{ AND } A1_{in}\ \theta\ A2_{jn}$$

Para facilitar la notación, sin perder por ello generalidad, podemos suponer que la condición de combinación se refiere a un par de columnas (k y l) y la expresamos como $k\ \theta\ l$.

Formalmente la combinación de las dos relaciones de esquemas R1(A1) y R2(A2) respecto de sus columnas k y l, denotada:

$$\underset{k\ \theta\ l}{R_1|X|R_2}$$

Es otra relación de grado n_1+n_2 cuyo esquema R estará formado por los n_1+n_2 atributos: A1 A2, es decir $(A1_1,\ldots,A1_{n1},A2_1,\ldots,A2_{n2})$, y cuya extensión r(R), de cardinalidad $\leq m_1 \times m_2$ será:

$$\{<V1_{i1} \ldots V1_{in1}\ V2_{j1} \ldots V2_{jn2}> / \forall i\ \forall j\ (<V1_{i1} \ldots V1_{in1}> \in r(R1) \wedge <V2_{j1} \ldots V2_{jn2}> \in r(R2) \wedge V1_{ik}\ \theta\ V2_{jl}\}$$

Si se trata de una condición de combinación simple por igualdad se denomina equi-combinación. La llamada combinación natural (a la que denotaremos por *) es una combinación por igualdad donde se ha eliminado, en la relación resultante, uno de los dos atributos cuyos valores son idénticos. Es el caso más utilizado de combinación para relaciones que tienen un atributo común.

En la figura 2.3.7.1 puede verse un ejemplo de aplicación del operador combinación natural.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

ASIGNATURA

NUMERO	DATOS	PROFESOR
10000	Tratamiento de datos	H0001
10001	Análisis estadístico	H0001
10002	Cálculo numérico	H0002

PROFESOR * ASIGNATURA
(PROFESOR.CODIGO=ASIGNATURA.PROFESOR)

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	DATOS
H0001	Antonio	García García	Catedrático	10000	Tratamiento de datos
H0001	Antonio	García García	Catedrático	10001	Análisis estadístico
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico

Figura 2.3.7.1: Ejemplo de aplicación del operador combinación natural.

Como puede observarse, la combinación es un producto cartesiano seguido de restricción, y la combinación natural es un producto cartesiano seguido de una restricción por igualdad y de proyección.

2.3.8 El operador derivado binario intersección (∩).

La intersección de dos relaciones R1 y R2 compatibles en su esquema es otra relación definida sobre el mismo esquema de relación y cuya extensión estará constituida por las tuplas que pertenezcan a ambas relaciones.

Formalmente, sean dos relaciones compatibles con esquemas R1 y R2, la intersección de ambas, denotada por será una relación r con esquema R (igual a R1 o a R2 ya que ambos son equivalentes y con extensión:

$$r(R) = \{t_i \ / \ t_i \in r_1 \wedge t_i \in r_2\}$$

En la figura 2.3.8.1 puede verse un ejemplo de aplicación del operador intersección.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular

PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

PROFESOR ∩ PDI

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0003	Isabel	Fernández Fernández	Titular

Figura 2.3.8.1: Ejemplo de aplicación del operador intersección.

La intersección se puede definir en función de la diferencia como R1 ∩ R2 = R1 – (R1 – R2) o también R2 – (R2 – R1).

2.3.9 El operador derivado binario división (:).

La división de una relación R1 (dividendo) por otra R2 (divisor) es una relación R (cociente) tal que, al realizarse su combinación con el divisor, todas las tuplas resultantes se encuentran en el dividendo.

Formalmente, sean dos relaciones con esquemas R1(A1) y R2(A2) de grados n_1 y n_2, respectivamente, donde A2 ⊂ A1 y, por tanto, $n_1 < n_2$. La división de ambas relaciones, denotada por R1 : R2 será una relación r de grado n_1-n_2, cuyo esquema R estará formado por los n_1-n_2 atributos A1 – A2:

$$(A1_1, A1_2, \ldots, A1_{n1}) - (A2_1, A2_2, \ldots, A2_{n2})$$

Y cuya extensión será:

$$r(R) = \{<V_{i1}, \ldots, V_{i(n1-n2)}> / \ \forall <V_{i(n1-n2+1)}, \ldots, V_{in2}> \in r_2 \ \exists <V_{i1}, \ldots, V_{i(n1-n2)}, V_{i(n1-n2+1)}, \ldots, V_{in}> \in r_1\}$$

En la figura 2.3.9.1 puede verse un ejemplo de aplicación del operador división.

PROFESOR_ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	ASIGNATURA
H0001	Antonio	García García	Catedrático	10000
H0001	Antonio	García García	Catedrático	10001
H0002	Amparo	Pérez Pérez	Ayudante	10000
H0002	Amparo	Pérez Pérez	Ayudante	10001
H0002	Amparo	Pérez Pérez	Ayudante	10002
H0003	Isabel	Fernández Fernández	Titular	10003
H0004	José	Hernández Hernández	Ayudante	10002
H0004	José	Hernández Hernández	Ayudante	10003
H0005	Carlos	Martínez Martínez	Titular	10000
H0005	Carlos	Martínez Martínez	Titular	10002

ASIGNATURA

ASIGNATURA
10000
10002

PROFESOR_ASIGNATURA : ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0002	Amparo	Pérez Pérez	Ayudante
H0005	Carlos	Martínez Martínez	Titular

Figura 2.3.9.1: Ejemplo de aplicación del operador división.

La división se puede expresar en función de la proyección, del producto cartesiano y de la diferencia de la siguiente forma:

$$R1 : R2 = \pi_C(R1) - \pi_C\ (R2\ X\ \pi_C\ (R1) - R1)$$

Siendo C el conjunto de atributos de A menos B.

2.3.10 Operadores adicionales de consulta.

A fin de mejorar el poder expresivo del álgebra relacional se han introducido ciertos operadores adicionales que no pueden considerarse como derivados dado que no es posible expresarlos en función de los operadores primitivos. Tales operadores son la *agrupación* (“group by”) y el *cierre transitivo*.

2.3.10.1 Agrupación.

Para aplicar funciones de agregación (frecuencia, suma, media, etc.), podemos agrupar tuplas en subconjuntos que posean valores comunes de ciertos atributos, como puede verse en la figura 2.3.10.1.1.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	SALARIO
H0001	Antonio	García García	Catedrático	3500
H0002	Amparo	Pérez Pérez	Ayudante	1500
H0003	Isabel	Fernández Fernández	Titular	2400
H0004	José	Hernández Hernández	Ayudante	1600
H0005	Carlos	Martínez Martínez	Titular	2600

PROFESOR AGRUPACION_POR CATEGORIA,MEDIA(SALARIO)

CATEGORIA	MEDIA(SALARIO)
Catedrático	3500
Ayudante	1550
Titular	2500

Figura 2.3.10.1.1: Ejemplo de operación de agrupación.

2.3.10.2 Cierre transitivo.

Es una operación unaria definida sobre dos atributos compatibles (comparten el mismo dominio) de una misma relación, que se obtiene por sucesivas operaciones de combinación, proyección y unión, consistente en añadir a la relación de origen todas las tuplas que se deducen, sucesivamente, por transitividad hasta la saturación; es decir, si existen las tupla (a,b) y (b,c) se añade (a,c). En la figura 2.3.10.2.1 puede verse un ejemplo de operación de cierre transitivo.

TEMAS

TEMA_P	TEMA_S
Física	Química
Química	Analítica
Química	Estadística
Matemáticas	Estadística
Estadística	Calculo

Cierre transitivo de TEMAS

TEMA_P	TEMA_S
Física	Química
Química	Analítica
Química	Estadística
Matemáticas	Estadística
Estadística	Calculo
Física	Analítica
Física	Estadística
Química	Calculo
Matemáticas	Calculo
Física	Calculo

Figura 2.3.10.2.1: Ejemplo de operación de cierre transitivo.

El cierre transitivo no se puede considerar un operador derivado, ya que no se conoce el número de operaciones de combinación, proyección y unión necesarias, no pudiéndose transformar, por tanto, en una única expresión del álgebra relacional; únicamente con el apoyo de un lenguaje anfitrión en el que se programara el correspondiente bucle se podría hacer la consulta.

2.3.11 Operadores relacionales con valores nulos.

Como señalamos anteriormente, los operadores nulos obligan a definir operaciones aritméticas, de comparación, algebraicas, etc., adicionales.

El producto cartesiano no se ve afectado por la presencia de valores nulos; la restricción sólo devuelve aquellas tuplas cuya condición evalúa como cierta, eliminando aquellas otras en las que el resultado de evaluar la condición es falso o quizás, y la proyección debe eliminar tuplas duplicadas teniendo en cuenta los nulos.

Veremos a continuación dos ejemplos de operadores con valores nulos, la *combinación externa* y los *operadores "posible" (maybe)*.

2.3.11.1 Combinación externa.

El operador combinación externa (outer join) es un operador que impide que desaparezcan tuplas cuando se aplica la combinación interna que hemos visto con anterioridad.

Así, por ejemplo, en la figura 2.3.7.1 donde se combinan las tablas PROFESOR y ASIGNATURA, las tuplas <H0003, Isabel, Fernández Fernández, Titular> <H0004,

José, Hernández Hernández, Ayudante> y <H0005, Carlos, Martínez Martínez, Titular> no aparecen en la relación resultante al no combinarse con ninguna tupla de la relación ASIGNATURA, es decir, se pierde esta información.

Para evitar que las tuplas de una relación que no casan con ninguna tupla de la otra desaparezcan en el resultado, podemos aplicar el operador de combinación externa que concatena esas tuplas con tuplas cuyos atributos tienen valores nulos, obteniéndose así, para el ejemplo anterior, el resultado que aparece en la figura 2.3.11.1.1.

PROFESOR

CODIGO	NOMBRE	APELLIDOS	CATEGORIA
H0001	Antonio	García García	Catedrático
H0002	Amparo	Pérez Pérez	Ayudante
H0003	Isabel	Fernández Fernández	Titular
H0004	José	Hernández Hernández	Ayudante
H0005	Carlos	Martínez Martínez	Titular

ASIGNATURA

NUMERO	DATOS	PROFESOR
10000	Tratamiento de datos	H0001
10001	Análisis estadístico	H0001
10002	Cálculo numérico	H0002

PROFESOR /* ASIGNATURA
(PROFESOR.CODIGO=ASIGNATURA.PROFESOR)

PROFESOR_ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	DATOS
H0001	Antonio	García García	Catedrático	10000	Tratamiento de datos
H0001	Antonio	García García	Catedrático	10001	Análisis estadístico
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico
H0003	Isabel	Fernández Fernández	Titular	NULL	NULL
H0004	José	Hernández Hernández	Ayudante	NULL	NULL
H0005	Carlos	Martínez Martínez	Titular	NULL	NULL

Figura 2.3.11.1.1: Ejemplo de aplicación del operador de combinación externa.

En este caso basta con aplicar el operador de combinación externa izquierda (representado por /*), ya que se desean guardar todas las tuplas de la tabla que está a la izquierda, siendo en la segunda relación donde hay que crear tuplas nulas.

Para el caso simétrico, esto es, cuando se desea que no se pierda ninguna tupla de la tabla que está a la derecha, será necesario crear tuplas en la primera tabla, utilizándose el operador de combinación externa derecha (representado por */). Puede en algún caso ser necesario crear tuplas nulas en ambas relaciones, denominándose en este caso el operador combinación externa simétrica (representado por /*/).

2.3.11.2 Operadores “posible” (maybe).

Otro tipo de operadores que tienen en cuenta los valores nulos son los denominados “posibles” (maybe), los cuales utilizan la lógica trivaluada y admiten como tuplas resultantes no sólo las que cumplen el valor cierto en una comparación, sino también las que cumplen puede ser cierto. Por ejemplo, en la figura 2.3.11.2.1 se presenta el caso de una combinación por igualdad de las relaciones R1 y R2, resultando la relación R3. Ahora bien, aplicando la combinación “posible”, el número de tuplas resultante es considerablemente superior, ya que la primera tupla de la relación A<1,5> no sólo se combina con la tupla <5,3> de la relación R2, sino que también se combina con la <”nulo, 6> y <”nulo, 7>.

R1

a	**b**
1	5
1	Nulo
2	4
Nulo	2
3	Nulo

R2

c	**d**
5	3
2	4
Nulo	6
Nulo	7

$R1 \bowtie_{b=c} R2$

a	**b**	**c**	**d**
1	5	5	3
Nulo	2	2	4

MAYBE $R1 \bowtie_{b=c} R2$

a	**b**	**c**	**d**
1	5	5	3
1	5	Nulo	6
1	5	Nulo	7
1	Nulo	5	3
1	Nulo	2	4
1	Nulo	Nulo	6
1	Nulo	Nulo	7
2	4	Nulo	6
2	4	Nulo	7
Nulo	2	2	4
Nulo	2	Nulo	6
Nulo	2	Nulo	7
3	Nulo	5	3
3	Nulo	2	4
3	Nulo	Nulo	6
3	Nulo	Nulo	7

Figura 2.3.11.2.1: Ejemplo de combinación posible simétrica.

El operador “maybe” se puede aplicar con cualquier otro operador donde pueda producirse una comparación en la que intervengan valores nulos; por ejemplo con la restricción. Así en la figura 2.3.11.2.2 se representa el resultado de aplicar el operador “maybe” al resultado de la operación de combinación externa de la figura 2.3.11.2.2.

PROFESOR_ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	DATOS
H0001	Antonio	García García	Catedrático	10000	Tratamiento de datos
H0001	Antonio	García García	Catedrático	10001	Análisis estadístico
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico
H0003	Isabel	Fernández Fernández	Titular	Nulo	Nulo
H0004	José	Hernández Hernández	Ayudante	Nulo	Nulo
H0005	Carlos	Martínez Martínez	Titular	Nulo	Nulo

$\sigma_{NUMERO=\text{"10002"}}$PROFESOR_ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	DATOS
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico

MAYBE($\sigma_{NUMERO=10002}$PROFESOR_ASIGNATURA

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	NUMERO	DATOS
H0002	Amparo	Pérez Pérez	Ayudante	10002	Cálculo numérico
H0003	Isabel	Fernández Fernández	Titular	Nulo	Nulo
H0004	José	Hernández Hernández	Ayudante	Nulo	Nulo
H0005	Carlos	Martínez Martínez	Titular	Nulo	Nulo

Figura 2.3.11.2.2: Ejemplo de aplicación del operador “maybe” sobre la operación restricción.

2.4 Ejercicios de álgebra relacional.

Dadas las relaciones mostradas a continuación:

AULAS	
COD_AULA	*CAPACIDAD*
A100	120
A110	100
A120	90
A130	90

ESTUDIANTES		
COD_ESTUD	*APELLIDOS*	*NOMBRE*
E0000	Abellan Andrés	Juan José
E0001	Benito Lafuente	Ana María
E0002	Botella Rocamora	Paloma María
E0003	Bustamante García	Elíseo
E0004	Candel Ayala	Roberto
E0005	Company Boronat	Asunción
E0006	Daza Arboli	Miguel Ángel
E0007	Abellan Andrés	Carlos
E0008	Agost Porcar	Ana
E0009	Alacreu García	Mónica
E0010	Alemany Llopis	Francisca
E0011	Argente Andrino	Estela
E0012	Aroco Belmonte	Eva

MATRICULA		
COD_ESTUD	*COD_ASIG*	*CALIFICACION*
E0000	10000	Aprobado
E0000	10003	Suspenso
E0000	10006	NULL
E0001	10004	Sobresaliente
E0001	10007	NULL
E0002	10004	Aprobado
E0003	10001	Aprobado
E0003	10002	Suspenso
E0004	10003	No Presentado
E0004	10008	Notable
E0004	10009	NULL
E0005	10005	Sobresaliente
E0005	10008	NULL
E0006	10006	Aprobado
E0007	10004	Notable
E0007	10007	Suspenso
E0008	10005	Suspenso
E0009	10001	Notable
E0009	10002	Aprobado
E0010	10004	NULL
E0011	10007	No Presentado
E0012	10000	No Presentado
E0012	10009	NULL

ASIGNATURAS		
COD_ASIG	*NOMBRE*	*CURSO*
10000	Adquisición y tratamiento...	1
10001	Análisis estadístico de datos	1
10002	Cálculo numérico y...	1
10003	Diseño de experimentos	1
10004	Economía	2
10005	Ecuaciones funcionales	2
10006	Inferencia y decisión	2
10007	Modelos estocásticos de la...	3
10008	Probabilidad y procesos...	3
10009	Programación matemática	3

HORARIOS		
COD_ASIG	*COD_AULA*	*HORARIO*
10000	A100	09:00-10:00
10001	A100	10:30-11:30
10002	A110	11:30-12:30
10003	A110	12:30-13:30
10004	A120	16:00-17:00
10005	A120	17:00-18:00
10006	A120	18:00-19:00
10007	A130	10:30-11:30
10008	A130	11:30-12:30
10009	A130	12:30-13:30

Contestar a las siguientes preguntas, utilizando para ello los operadores del álgebra relacional.

1) Código y capacidad de las aulas con capacidad para 100 o más alumnos.

2) Código y calificación de los alumnos que estudian la asignatura de código 10004.

3) Nombre y apellidos de los alumnos que estudian la asignatura de código 10005.

4) Horario del alumno de código E0004.

5) Capacidad de las aulas en las que asiste a clase el estudiante de código E0000.

6) Nombre y apellidos de los alumnos que figuran en la matrícula de la asignatura “Diseño de experimentos”

7) Código de los alumnos que tienen todas las asignaturas de las que figuran matriculados en el mismo curso.

8) Nombre y apellidos de los estudiantes que tienen al menos una asignatura aprobada (“Aprobado”, “Notable” o “Sobresaliente”).

9) Código de los estudiantes y las asignaturas pendientes de aprobar.

10) Código de los estudiantes que con total seguridad tienen todas las asignaturas de las que están matriculados aprobadas.

11) Nombre y apellidos de los estudiantes que pueden tener todas las asignaturas aprobadas.

12) Nombre y apellidos de los estudiantes que tienen un hermano estudiando.