

Sistemas y Aplicaciones  
Informáticas

Tema 37. Modelo de Datos  
Relacional. Estructura. Operaciones.  
Álgebra Relacional.

<b>1. ÁMBITO DE DOCENCIA.</b> .....	<b>3</b>
<b>2. MODELO DE DATOS RELACIONAL</b> .....	<b>3</b>
2.1. MODELOS DE DATOS. ESQUEMAS. DISEÑO DE BASES DE DATOS. ....	3
2.2. MODELO RELACIONAL. ....	3
2.2.1. <i>Concepto. Relaciones.</i> .....	3
2.2.2. <i>Estructura del modelo relacional.</i> .....	4
2.2.2.1. Relaciones. Atributos y grado. Tuplas y cardinalidad. Valores nulos. ....	4
2.2.2.2. Dominios de los atributos. Dominios simples y compuestos. ....	4
2.2.2.3. Claves candidatas y propiedades. Claves primarias, alternativas y ajenas. ....	4
2.2.2.4. Reglas de integridad del modelo relacional. ....	5
2.2.3. <i>Propiedades y tipos de relaciones.</i> .....	5
2.2.4. <i>El modelo relacional según la arquitectura ANSI.</i> ....	6
2.3. REGLAS DE CODD PARA LOS DBMS RELACIONALES. ....	6
2.4. NORMALIZACIÓN Y FORMAS NORMALES. ....	7
2.4.1. <i>Normalización. Dependencia funcional y determinante. Formas normales.</i> ....	7
2.4.2. <i>Primera forma normal (1FN).</i> ....	7
2.4.3. <i>Segunda forma normal (2FN).</i> ....	8
2.4.4. <i>Tercera forma normal (3FN).</i> ....	8
2.4.5. <i>Forma normal de Boyce-Codd (BCFN).</i> ....	8
<b>3. ÁLGEBRA RELACIONAL. OPERACIONES.</b> .....	<b>8</b>
3.1. CONCEPTO DE ÁLGEBRA RELACIONAL. OPERACIONES. ....	8
3.2. DESCRIPCIÓN DE OPERACIONES BÁSICAS. ....	9
3.3. DESCRIPCIÓN DE OPERACIONES ADICIONALES. ....	10

## 1. **Ámbito de docencia.**

- Sistemas informáticos monousuario y multiusuario (ASI 1).
- Sistemas informáticos multiusuario y en red (DAI 1).
- Instalación y mantenimiento de equipos y sistemas informáticos (ESI 1).

## 2. **Modelo de datos relacional.**

### 2.1. **Modelos de datos. Esquemas. Diseño de bases de datos.**

- Un modelo de datos es una colección de conceptos para la descripción de los datos, las relaciones entre ellos y las restricciones que deben cumplir. Un esquema es una descripción de una base de datos mediante un modelo de datos. Los modelos de datos se clasifican en:
  - \* *Modelos conceptuales.* Describen los datos con un alto nivel de abstracción, utilizando entidades (concepto del mundo real), atributos (propiedad de interés de una entidad) y relaciones (interacción entre dos o más entidades). Son independientes de la base de datos a utilizar. Por ejemplo, el modelo entidad/relación y el modelo orientado a objetos.
  - \* *Modelos lógicos.* Representan los datos valiéndose de estructuras de registros de varios tipos, formados por un número determinado de campos. Son dependientes de la base de datos a utilizar. Por ejemplo, el modelo relacional, el modelo de red y el modelo jerárquico.
  - \* *Modelos físicos.* Los modelos físicos describen cómo se almacenan los datos en cuanto al formato de los registros, la estructura de los ficheros y los métodos de acceso utilizados.
- El diseño de bases de datos se estructura en tres pasos:
  - \* *Diseño conceptual.* Recibe como entrada la especificación de requerimientos y su resultado es el esquema conceptual, que es una descripción de alto nivel de la estructura de la base de datos mediante un modelo conceptual, y que es independiente del DBMS que se utilice.
  - \* *Diseño lógico.* Recibe como entrada el esquema conceptual y da como resultado un esquema lógico, que es una descripción de la estructura de la base de datos mediante un modelo lógico, y que puede ser procesado por el DBMS que se utilice.
  - \* *Diseño físico.* Recibe como entrada el esquema lógico y da como resultado un esquema físico, que es una descripción mediante un modelo físico de las estructuras de almacenamiento y de los métodos usados para tener un acceso efectivo a los datos.

### 2.2. **Modelo relacional.**

#### 2.2.1. **Concepto. Relaciones.**

- El modelo relacional es un modelo lógico de datos que se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. En él, todos los datos están estructurados en los niveles externo y conceptual de la arquitectura ANSI como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta.
- El modelo relacional se basa en el concepto matemático de relación. Las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Los datos y las relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una con un nombre que es único.

## 2.2.2. Estructura del modelo relacional.

### 2.2.2.1. Relaciones. Atributos y grado. Tuplas y cardinalidad. Valores nulos.

- Una relación se representa gráficamente como una tabla bidimensional formada por:
  - \* *Atributos*. Es el nombre de una columna de una relación. El grado de una relación es el número de atributos que contiene. El grado de una relación no cambia con frecuencia.
  - \* *Tuplas*. Es una fila de una relación. Los elementos de una relación son las tuplas de la tabla. La cardinalidad de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente.
- Cuando en una tupla un atributo es desconocido, se dice que es nulo. El nulo implica ausencia de información, bien porque se desconoce su valor, o bien porque para dicha tupla el atributo no tiene sentido. Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

### 2.2.2.2. Dominios de los atributos. Dominios simples y compuestos.

- Un dominio es el conjunto de valores del mismo tipo que puede tomar un atributo. Cada atributo de una base de datos relacional se define sobre un único dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.
- Un dominio está formado por un conjunto de valores escalares, es decir, datos individuales sin estructura interna. Un dominio simple es un dominio de valores escalares. Un dominio compuesto es una combinación de dominios simples. Esto implica que los atributos podrán ser simples o compuestos, dependiendo de la naturaleza del dominio sobre el que estén definidos.
- El concepto de dominio permite que el usuario defina el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas se pueden evitar.
- Al especificar el dominio de un atributo, implícitamente se indican los operadores unarios y binarios que pueden aplicarse a los valores del atributo. Además permite especificar para cada expresión escalar legal, el dominio del resultado de evaluar dicha expresión.

### 2.2.2.3. Claves candidatas y propiedades. Claves primarias, alternativas y ajenas.

- Puesto que en una relación no puede haber tuplas repetidas, es posible identificarlas de modo único mediante los valores de sus atributos. Una superclave es un atributo o un conjunto de atributos que identifican de modo único las tuplas de una relación. Una clave candidata es una superclave en la que ninguno de sus subconjuntos es una superclave de la relación.
- Cuando una clave candidata está formada por más de un atributo, se dice que es una clave compuesta. Una relación puede tener varias claves candidatas. El atributo o conjunto de atributos  $K$  de la relación  $R$  es una clave candidata para  $R$  si y sólo si satisface las siguientes propiedades:
  - \* *Unicidad*. Nunca hay dos tuplas en la relación  $R$  con el mismo valor de  $K$ .
  - \* *Minimalidad*. Ningún subconjunto de  $K$  sirve para identificar las tuplas de modo único.
- La clave primaria de una relación es aquella clave candidata que se escoge para identificar sus tuplas de modo único. Ya que una relación no tiene tuplas duplicadas, siempre hay una clave

candidata y por lo tanto, la relación siempre tiene clave primaria. Las claves candidatas que no son escogidas como clave primaria son denominadas claves alternativas.

- Una clave ajena es un atributo o un conjunto de atributos de una relación cuyos valores coinciden con los valores de la clave primaria de alguna otra relación. Las claves ajenas representan relaciones entre los datos de tablas padres e hijas. Se dice que un valor de clave ajena de una tabla hija representa una referencia a la tupla de la tabla padre que contiene el mismo valor en su clave primaria. Por tanto, ambas deben definirse sobre el mismo dominio.

#### 2.2.2.4. Reglas de integridad del modelo relacional.

- **Regla de integridad de entidades.** Ninguno de los atributos que componen la clave primaria puede ser nulo. Por definición, ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, lo que contradice la propiedad de minimalidad. Esta regla sólo se aplica a las claves primarias, no a las claves alternativas.
- **Regla de integridad referencial.** Si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que referencia, o bien deben ser nulos:
  - \* Si se intenta borrar la tupla referenciada por la clave ajena, o modificar el valor de su clave primaria, se puede realizar una de las siguientes acciones:
    - **Restringir (RESTRICT).** No se permite borrar la tupla referenciada, o modificar el valor de su clave primaria.
    - **Propagar (CASCADE).** Se borra la tupla referenciada, o se modifica el valor de su clave primaria, y se propaga el borrado o la modificación a las tuplas que la referencian mediante la clave ajena.
    - **Anular (SET NULL).** Se borra o se modifica la tupla referenciada y las tuplas que hacían referencia a ella ponen a nulo la clave ajena (sólo si acepta nulos).

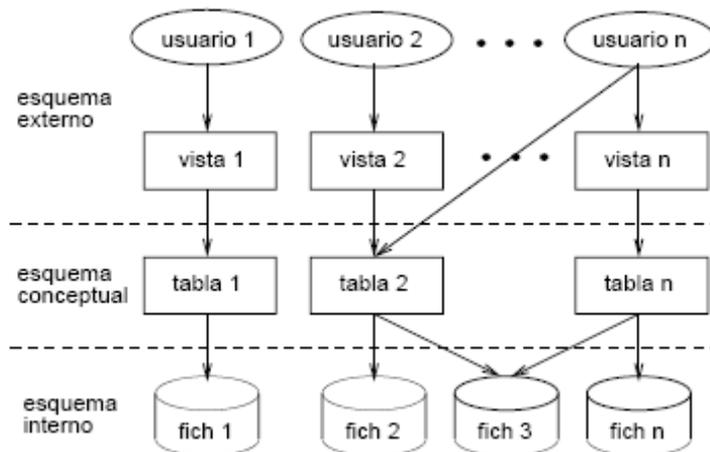
#### 2.2.3. Propiedades y tipos de relaciones.

- Las relaciones tienen las siguientes propiedades:
  - \* Cada relación y cada atributo de una misma relación tienen nombres distintos.
  - \* En cada tupla, un atributo toma un solo valor. No existen tuplas duplicadas.
  - \* El orden de los atributos y las tuplas dentro de las tablas es indiferente.
- En un DBMS relacional pueden existir varios tipos de relaciones:
  - \* *Relaciones base.* Son relaciones reales con nombre que forman parte de la base de datos.
  - \* *Vistas.* También denominadas relaciones virtuales, son relaciones derivadas con nombre. Se definen en función de otras relaciones con nombre, no poseen datos almacenados propios.
  - \* *Instantáneas.* Son relaciones derivadas con nombre reales, no virtuales. Se definen en función de otras relaciones con nombre, aunque poseen datos almacenados propios. Son relaciones de sólo lectura y se refrescan periódicamente.
  - \* *Resultados de consultas.* Son las relaciones resultantes de alguna consulta especificada. Pueden o no tener nombre y no persisten en la base de datos.
  - \* *Resultados intermedios.* Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.

- \* *Resultados temporales.* Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento.

#### 2.2.4. El modelo relacional según la arquitectura ANSI.

- El esquema conceptual de una base de datos relacional está formado por el nombre de sus relaciones y sus atributos, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajenas.
- El esquema externo está formado por las vistas o relaciones virtuales que se definen sobre una o más relaciones base.
- El esquema interno no está definido en el modelo relacional puesto que se trata de un modelo lógico. Cada producto ofrecerá los elementos internos como índices, agrupamientos, tablas hash, particiones, etc, que el administrador utilizará para mejorar el rendimiento del sistema.



#### 2.3. Reglas de Codd para los DBMS relacionales.

- Codd propuso en 1985 un conjunto de doce reglas para certificar que un DBMS es relacional:
  - \* *Representación de la información.* Toda información en una base de datos relacional debe presentarse explícitamente a nivel lógico, y de manera única, por medio de valores en tablas.
  - \* *Acceso por clave primaria.* Todo dato debe ser accesible mediante una combinación de un nombre de tabla, un valor de su clave primaria y el nombre de una columna.
  - \* *Tratamiento sistemático de valores nulos.* Los valores nulos han de ser tratados sistemáticamente por el sistema, ofreciendo las facilidades necesarias para su tratamiento.
  - \* *Diccionario de datos basado en el modelo relacional.* La representación de la descripción de la base de datos debe ser igual a la de los otros datos, y su acceso debe realizarse por medio del mismo lenguaje relacional que se utiliza para los demás datos.
  - \* *Lenguaje de datos completo.* Debe existir un lenguaje que permita un completo manejo de la base de datos (definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorizaciones y gestión de transacciones).
  - \* *Actualización de vistas.* Toda vista actualizable debe poder ser actualizada por el sistema.
  - \* *Inserciones, modificaciones y eliminaciones de alto nivel.* Todas las operaciones de manipulación de datos (consulta, inserción, modificación y borrado) deben operar sobre conjuntos de filas utilizando un lenguaje no navegacional y no procedimental.

- \* *Independencia física de los datos.* El acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento.
  - \* *Independencia lógica de los datos.* Los programas de aplicación no deben verse afectados por cambios realizados en las tablas que estén permitidos y que preserven la información.
  - \* *Independencia de la integridad.* Las reglas de integridad de una base de datos deben ser definibles por medio del sublenguaje de datos relacional y habrán de almacenarse en el diccionario de la base de datos, no en los programas de aplicación.
  - \* *Independencia de la distribución.* Debe existir un sublenguaje de datos que pueda soportar bases de datos distribuidas, sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o se redistribuyan éstos posteriormente.
  - \* *Regla de la no subversión.* Si un SGBD soporta un lenguaje de bajo nivel que permite el acceso fila a fila, éste no puede utilizarse para saltarse las reglas de integridad expresadas por medio del lenguaje de más alto nivel.
- Estas reglas pueden resumirse en que todos los datos deben estar organizados en tablas y que todas las operaciones deben actuar sobre dichas tablas.

## **2.4. Normalización y formas normales.**

### **2.4.1. Normalización. Dependencia funcional y determinante. Formas normales.**

- La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional. Es una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimina las dependencias no deseadas entre atributos. Una base de datos relacional es un conjunto de relaciones normalizadas.
- Una dependencia funcional es una relación entre atributos de una misma relación (tabla). Si  $x$  e  $y$  son atributos de la relación  $R$ , se dice que  $y$  es funcionalmente dependiente de  $x$  (se denota por  $x \rightarrow y$ ) si cada valor de  $x$  tiene asociado un solo valor de  $y$  ( $x$  e  $y$  pueden constar de uno o varios atributos). A  $x$  se le denomina determinante, ya que  $x$  determina el valor de  $y$ . Se dice que el atributo  $y$  tiene dependencia funcional completa de  $x$  si depende funcionalmente de  $x$  y no depende de ningún subconjunto de  $x$ .
- La normalización se basa en las formas normales. Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones. Si éstas no se cumplen, la relación se debe descomponer en varias relaciones que sí las cumplan.
- Mediante la normalización, una relación en una determinada forma normal se convierte en un conjunto de relaciones en una forma normal superior, siendo un proceso reversible. Por definición, el modelo relacional se encuentra en primera forma normal. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

### **2.4.2. Primera forma normal (1FN).**

- Una relación está en 1FN si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, si tiene un solo valor en la intersección de cada fila con cada columna.
- Si una relación no está en 1FN, hay que eliminar de ella el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos maneras:

- \* Repetir los atributos con un solo valor para cada uno de los valores múltiples. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales.
- \* Poner cada uno de los atributos o los grupos de atributos que tiene múltiples valores en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

#### **2.4.3. Segunda forma normal (2FN).**

- Una relación está en 2FN si y sólo si está en 1FN, y además cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.
- Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones. Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de los atributos de la clave primaria de los que dependen.

#### **2.4.4. Tercera forma normal (3FN).**

- Una relación está en 3FN si y sólo si está en 2FN y además cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia  $x \rightarrow z$  es transitiva si existen las dependencias  $x \rightarrow y$ ,  $y \rightarrow z$  siendo  $x$ ,  $y$ ,  $z$  atributos o conjuntos de atributos de una misma relación.
- Las relaciones en 2FN todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de los atributos no clave de los que dependen.

#### **2.4.5. Forma normal de Boyce-Codd (BCFN).**

- La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si éstas existen. La BCFN es más fuerte que la 3FN, por tanto, toda relación en BCFN está en 3FN.
- Una relación está en BCFN si y sólo si todo determinante es una clave candidata. Dicho de otro modo, la definición implica que los únicos determinantes son las claves candidatas. Se debe comprobar si una relación viola la BCFN si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común.

### **3. Álgebra relacional. Operaciones.**

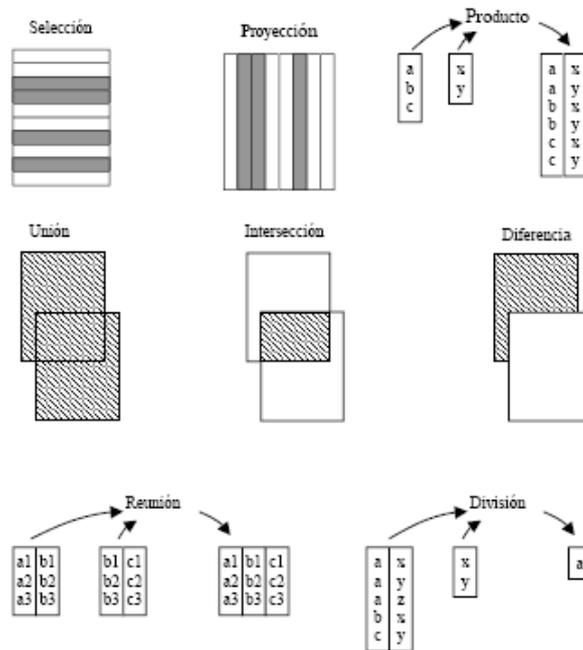
#### **3.1. Concepto de álgebra relacional. Operaciones.**

- El álgebra relacional es un lenguaje procedimental de alto nivel con una serie de operadores que trabajan sobre una o varias relaciones para obtener otra relación resultado, sin modificar las relaciones originales, por lo que la salida de una operación puede ser la entrada de otra operación, permitiendo el anidamiento de expresiones.
- Existen ocho operaciones básicas, de los cuales cinco son fundamentales: restricción, proyección, producto cartesiano, unión y diferencia, que permiten realizar la mayoría de las operaciones de obtención de datos. Las operaciones no fundamentales son la reunión, la intersección y la división, que se pueden expresar a partir de los cinco operadores fundamentales.

### 3.2. Descripción de operaciones básicas.

- Las operaciones básicas fundamentales del álgebra relacional son las siguientes:
  - \* **Restricción (R WHERE condición).** También denominada selección, opera sobre una sola relación R y da como resultado otra relación cuyas tuplas son las tuplas de R que satisfacen la condición especificada. Esta condición es una comparación en la que aparece al menos un atributo de R, o una combinación booleana de varias de estas comparaciones. Una selección sobre R se puede escribir como  $\sigma_{\text{campo}=\text{"dato"}}(R)$ , cuyo resultado es una relación que contiene aquellas tuplas de R que cumplen que el atributo "campo" tiene el valor "dato". Es posible utilizar en el predicado los operadores  $\leq, \geq, \neq, <, >, =$ .
  - \* **Proyección (R [a<sub>1</sub>, ..., a<sub>k</sub>]).** Opera sobre una sola relación R y da como resultado otra relación que contiene un subconjunto vertical de R, extrayendo los valores de los atributos especificados. Una proyección sobre R se puede escribir como  $\pi_{c1,c2,c3}(R)$ , cuyo resultado es una relación formada por los atributos c1, c2 y c3 de R.
  - \* **Producto cartesiano (R TIMES S).** El producto cartesiano de dos relaciones R y S ( $R \times S$ ) obtiene una relación cuyas tuplas están formadas por todos los pares posibles de tuplas de las dos relaciones. Si la relación R tiene P tuplas y N atributos y la relación S tiene Q tuplas y M atributos, la relación resultado tendrá P\*Q tuplas y N+M atributos.
  - \* **Unión (R UNION S).** La unión de dos relaciones R y S ( $R \cup S$ ) es otra relación que tiene las tuplas que se encuentran en R o en S o en ambas relaciones a la vez. Para poder realizar esta operación, R y S deben ser compatibles para la unión, es decir, tener el mismo número de atributos y que éstos se encuentren definidos sobre los mismos dominios.
  - \* **Diferencia (R MINUS S).** La diferencia de dos relaciones R y S ( $R - S$ ) obtiene una relación que tiene las tuplas que se encuentran en R y no se encuentran en S. Para realizar esta operación, R y S deben ser compatibles para la unión.
- Las operaciones básicas no fundamentales del álgebra relacional son las siguientes:
  - \* **Intersección (R INTERSECT S).** La intersección de dos relaciones R y S ( $R \cap S$ ) obtiene una relación que contiene las tuplas de R que también se encuentran en S. Para realizar esta operación, R y S deben ser compatibles para la unión. La intersección se puede expresar en términos de diferencias:  $R \text{ INTERSECT } S = R \text{ MINUS } (R \text{ MINUS } S)$ .
  - \* **División (R DIVIDEBY S).** Suponiendo que la cabecera de R es el conjunto de atributos A y que la cabecera de S es el conjunto de atributos B, tales que B es un subconjunto de A, y si C está formado por los atributos de R que no están en S, la división de dos relaciones R y S ( $R \div S$ ) obtiene una relación cuya cabecera es el conjunto de atributos C y que contiene las tuplas de R que están acompañadas de todas las tuplas de S.
  - \* **Reunión (R JOIN S).** La reunión de dos relaciones R y S ( $R \bowtie S$ ) obtiene una relación cuyas tuplas están formadas por todos los pares de tuplas de las dos relaciones cuyos atributos comunes tienen los mismos valores. Estos atributos comunes aparecen una sola

vez en el resultado. La reunión es, en realidad, un producto cartesiano y una restricción de igualdad sobre los atributos comunes.



### 3.3. Descripción de operaciones adicionales.

- **Reunión externa ( $R \text{ JOIN } S(+)$ ).** La reunión externa de dos relaciones  $R$  y  $S$  es una reunión en la que las tuplas de  $R$  que no tienen valores en común con ninguna tupla de  $S$ , también aparecen en el resultado. La expresión  $S(+)$  JOIN  $R$  es equivalente a  $R \text{ JOIN } S(+)$ . Cuando en ambas relaciones hay tuplas que no se pueden concatenar y se desea que en el resultado aparezcan también todas estas tuplas (tanto las de una relación como las de la otra), se utiliza la reunión externa completa:  $R(+)$  JOIN  $S(+)$ .
- **Agrupación (SUMMARIZE  $R$  GROUPBY( $a_1, \dots, a_k$ ) ADD cálculo AS atributo).** También denominada resumen, esta operación agrupa las tuplas de  $R$  que tienen los mismos valores en los atributos especificados y realiza un cálculo sobre los grupos obtenidos. La relación resultado tiene como cabecera los atributos por los que se ha agrupado y el cálculo realizado, al que se da el nombre especificado en atributo. Los cálculos que se pueden realizar sobre los grupos de filas son: suma de los valores de un atributo ( $SUM(a_p)$ ), media de los valores de un atributo ( $AVG(a_p)$ ), máximo y mínimo de los valores de un atributo ( $MAX(a_p)$ ,  $MIN(a_p)$ ) y número de tuplas en el grupo ( $COUNT(*)$ ). La relación resultado tendrá tantas filas como grupos se hayan obtenido.