

FICHEROS Y BASES DE DATOS (E44)

3º INGENIERÍA EN INFORMÁTICA

Tema 8.

Elementos Básicos

- 1.- Ejemplo Introdutorio.
- 2.- Dominios.
- 3.- Relaciones.
- 4.- Bases de Datos Relacionales.

(Capítulo 11 del Date)

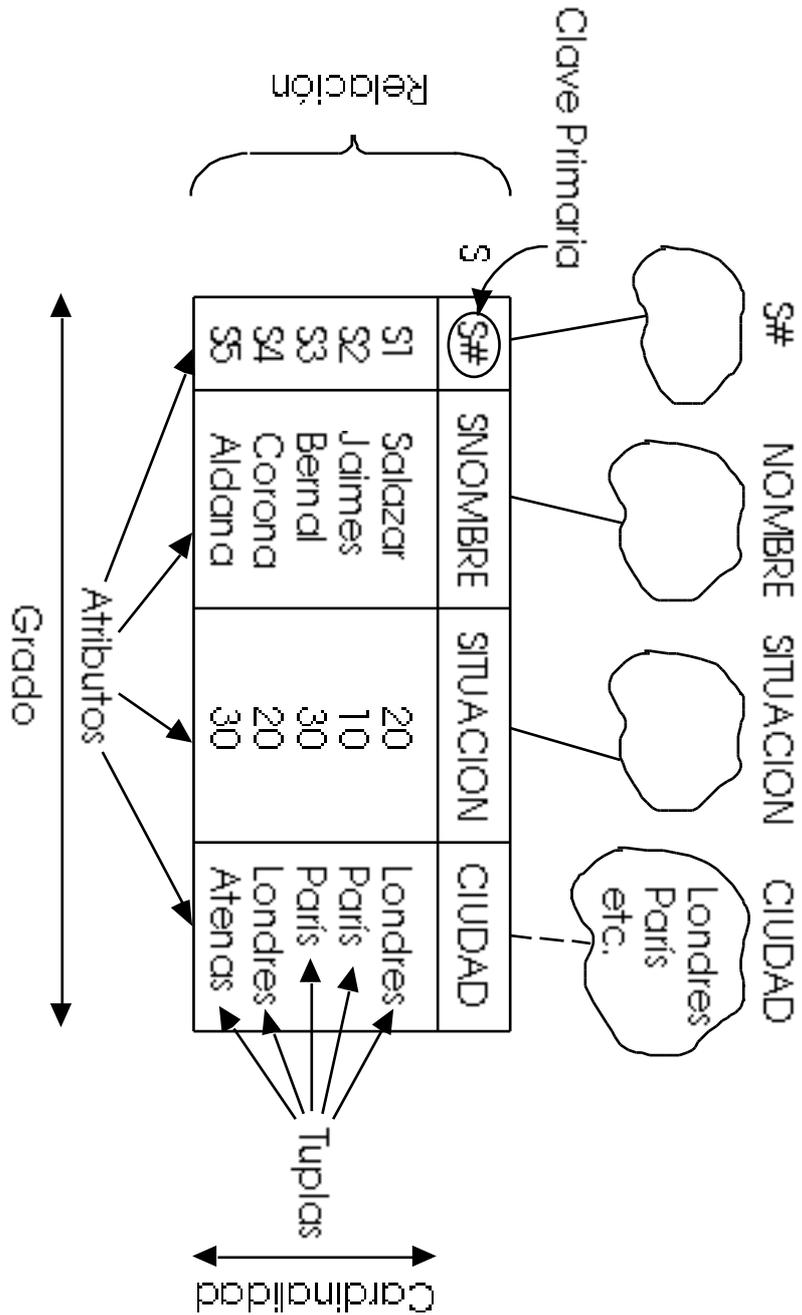
EJEMPLO INTRODUCTORIO

Definiciones

- Para describir la estructura de datos relacional se utiliza una terminología específica.
- Seguidamente se definen estos términos, mostrando algunos sinónimos más informales.
- Dichos sinónimos pueden no ajustarse de modo completo a la definición del término, pero son muy utilizados a nivel práctico.
- Relación (Tabla). Elemento básico del modelo relacional, que se asocia a una entidad.
- Tupla (Fila o Registro). Cada una de las instancias de una entidad.
- Cardinalidad (Número de Filas). Número de instancias de una entidad en una relación.
- Atributo (Campo o Columna). Cada una de las propiedades que caracterizan una entidad.
- Grado (Número de Columnas). Número de atributos asociados a una entidad.
- Clave Primaria (Identificador Único). Conjunto de uno o más atributos que identifican de modo completo una instancia de una entidad.
- Dominio (Conjunto de Valores Legales). Define la colección de valores de donde uno o más atributos obtienen sus valores reales.
- Algunos SGBD no permiten definir dominios asociados a los atributos, aunque el modelo relacional si contempla esta posibilidad.

EJEMPLO INTRODUCTORIO

Representación de las Definiciones



DOMINIOS

Definiciones

- En el análisis de la estructura relacional, se va a partir de los Escalares.
- Un escalar es la menor unidad semántica de información, es decir, es un dato atómico.
- La atomicidad se considera a nivel semántico, de modo que su descomposición produce una pérdida de significado.
- Se define un Dominio, o Dominio Simple, como un conjunto de escalares.

Características

- Cada atributo de una relación tiene asociado un Dominio Subyacente, de modo que el valor de las instancias del atributo se incluyen en él.
- No todos los valores del dominio subyacente deben aparecer como instancia del atributo.
- Diferentes atributos pueden tener asociado el mismo dominio subyacente.
- Los dominios permiten eliminar errores del programador, restringiendo la comparación a atributos asociados al mismo dominio.
- Esto se cumple también sobre dominios que sean equivalentes aunque semánticamente diferentes.
- En SQL no se incluye el concepto de dominio, permitiendo comparación entre atributos equivalentes.

DOMINIOS

Representación

- Los dominios se pueden especificar de modo explícito en la base de datos como un conjunto de valores, pero no es lo habitual.
- La alternativa más usual es incluirlos en la definición de la base de datos.
- Posteriormente, la definición de los atributos contiene el dominio subyacente.

Terminología

- Es aconsejable que el nombre de los dominios y de los atributos coincidan, siempre y cuando no exista ambigüedad.
- De este modo se puede omitir el dominio al que pertenece un atributo.

Consultas y Dominios

- Algunas consultas se pueden interpretar en el ámbito de los dominios.
- "Relaciones que contienen información sobre los proveedores".
- "Relaciones que incluyen un atributo definido en el dominio de los números de proveedor".
- Si existen los dominios, se realiza consultando los dominios en el catálogo del sistema.
- Si se cumple la norma terminológica anterior, se consulta el nombre de los atributos.

DOMINIOS

Ejemplo en Seudo-SQL

```
CREATE DOMAIN S#           CHAR(5);
CREATE DOMAIN NOMBRE      CHAR(20);
CREATE DOMAIN SITUACION   SMALLINT;
CREATE DOMAIN CIUDAD      CHAR(15);
CREATE DOMAIN P#          CHAR(6);
CREATE DOMAIN COLOR       CHAR(6);
CREATE DOMAIN PESO        SMALLINT;
CREATE DOMAIN CANT        INTEGER;

CREATE TABLE S
( S#           DOMAIN (S#)           NOT NULL,
  SNOMBRE     DOMAIN (NOMBRE)       NOT NULL,
  SITUACION   DOMAIN (SITUACION)    NOT NULL,
  CIUDAD      DOMAIN (CIUDAD)       NOT NULL,
  PRIMARY KEY (S#) );

CREATE TABLE P
( P#           DOMAIN (P#)           NOT NULL,
  PNOMBRE     DOMAIN (NOMBRE)       NOT NULL,
  COLOR       DOMAIN (COLOR)        NOT NULL,
  PESO        DOMAIN (PESO)         NOT NULL,
  CIUDAD      DOMAIN (CIUDAD)       NOT NULL,
  PRIMARY KEY (P#) );

CREATE TABLE SP
( S#           DOMAIN (S#)           NOT NULL,
  P#           DOMAIN (P#)           NOT NULL,
  CANT        DOMAIN (CANT)         NOT NULL,
  PRIMARY KEY (S#, P#),
  FOREIGN KEY (S#) REFERENCES S,
  FOREIGN KEY (P#) REFERENCES P );
```

DOMINIOS

Ejemplo en Seudo-SQL Omitiendo Dominios

```
CREATE DOMAIN S#           CHAR(5);
CREATE DOMAIN NOMBRE      CHAR(20);
CREATE DOMAIN SITUACION   SMALLINT;
CREATE DOMAIN CIUDAD      CHAR(15);
CREATE DOMAIN P#          CHAR(6);
CREATE DOMAIN COLOR       CHAR(6);
CREATE DOMAIN PESO        SMALLINT;
CREATE DOMAIN CANT        INTEGER;

CREATE TABLE S
( S#           NOT NULL,
  SNOMBRE     DOMAIN (NOMBRE) NOT NULL,
  SITUACION   NOT NULL,
  CIUDAD      NOT NULL,
  PRIMARY KEY (S#) );

CREATE TABLE P
( P#           NOT NULL,
  PNOMBRE     DOMAIN (NOMBRE) NOT NULL,
  COLOR       NOT NULL,
  PESO        NOT NULL,
  CIUDAD      NOT NULL,
  PRIMARY KEY (P#) );

CREATE TABLE SP
( S#           NOT NULL,
  P#           NOT NULL,
  CANT         NOT NULL,
  PRIMARY KEY (S#, P#),
  FOREIGN KEY (S#) REFERENCES S,
  FOREIGN KEY (P#) REFERENCES P );
```

DOMINIOS

Dominios Compuestos

- Un dominio compuesto es una combinación de dominios simples.
- Así, una instancia de un dominio compuesto es la asociación de una instancia de cada uno de los dominios simples que lo constituyen.
- Una vez definido se maneja como un dominio cualquiera.
- La Función Selectora que permite referenciar las componentes de una instancia de este tipo de dominios es el propio nombre de la componente.

Interpretación de los Dominios

- Desde un punto de vista intuitivo, un dominio es equivalente a un tipo de datos.
- De este modo, se define atributos (variables) que pertenezcan a un dominio (tipo de dato)
- No todos los lenguajes de alto nivel permiten la definición y el manejo de tipos de datos.
- De igual modo ocurre en los SGBD.
- En SQL se manejan los tipos básicos, pero el usuario no puede definir sus propios dominios.

DOMINIOS

Ejemplo de Dominios Compuesto en Seudo-SQL

```
CREATE DOMAIN DIA      CHAR(2);
CREATE DOMAIN MES      CHAR(2);
CREATE DOMAIN AÑO      CHAR(4);
CREATE DOMAIN FECHA    ( DIA DOMAIN (DIA)
                        MES  DOMAIN (MES)
                        AÑO  DOMAIN (AÑO));

CREATE TABLE EMPLEADO
  ( NUMEMP      DOMAIN (NUMEMP)  NOT NULL,
    .... ,
    FECHA_INGR  DOMAIN (FECHA)   NOT NULL,
    .... ,
    PRIMARY KEY (NUMEMP) );

SELECT ....
  FROM EMPLEADO
  WHERE EMPLEADO.FECHA_INGR = '01011993';

SELECT ....
  FROM EMPLEADO
  WHERE AÑO(EMPLEADO.FECHA_INGR) = '1993';
```

DOMINIOS

Problemas de los Dominios

- Su utilización es muy conveniente, pero se pueden presentar diferentes problemas.
- Las operaciones deben permitirse sobre dominios semejantes.
- No todas las comparaciones tienen sentido entre instancias de un dominio.
- Se debe permitir la combinación de dominios en las operaciones.

Manejo de los Dominios

- Se debe especificar el Conjunto Completo de Dominios, D , asociado a la base de datos.
- Dicho conjunto tiene que ser Cerrado, es decir, cualquier expresión escalar debe dar como resultado un escalar del conjunto.
- Para cada dominio se definen los Operadores Unarios que se pueden aplicar sobre sus instancias.
- Para cada combinación de $n = 2, 3, \dots$ dominios, se definen los Operadores n -arios que se pueden utilizar.
- También debe definirse, el dominio del resultado de cualquier expresión escalar legal.

RELACIONES

Definiciones

- Una Relación sobre un conjunto de dominios se compone de dos partes, la Cabecera y el Cuerpo.
- La Cabecera, o Esquema de la Relación, se compone de un conjunto fijo de pares atributo-dominio.
- El número de estos pares se denomina Grado, o Aridad, de la relación.
- El Cuerpo, o Extensión de la Relación, está formado por un conjunto Tuplas.
- El número de tuplas es la Cardinalidad de la Relación.
- Cada tupla se compone de un conjunto de pares atributo-valor.

Representación de las Relaciones

- Una relación tal y como se ha definido es un objeto bastante abstracto.
- Una tabla se puede entender como una representación de una relación en un instante.
- No todas las tablas representan una relación.
- Para que una tabla represente una relación debe de cumplir los enunciados anteriores.
- Por lo tanto se deben de convenir una serie de Reglas de Interpretación que permitan "leer" convenientemente la tabla.

RELACIONES

Interpretación de las Tablas

- Existe un dominio subyacente asociado a cada columna de la tabla.
- Dicho dominio no se muestra en las tablas, aunque de modo conceptual si aparece, ya que normalmente se nombran de igual modo.
- Las dos partes de la tabla corresponden a las dos partes de la relación.
- En la fila de cabecera de columnas aparecen los atributos de la relación, y por lo anterior, el dominio subyacente a cada uno.
- En el resto de la tabla aparece un conjunto de tuplas cuyo contenido varía en el tiempo.
- Las filas se forman de pares atributo-valor.
- Los valores incluidos en una fila se asocian al atributo de la columna a la que pertenecen, por lo que realmente son pares atributo-valor.

Tablas y Dominios

- Algunas veces se dice que un dominio se representa mediante una tabla 1-aria, pero es falso.
- La diferencia básica es que una tabla puede variar su contenido y un dominio es fijo.
- Si existen varias columnas asociadas al mismo dominio, éstas se nombran agregando al nombre del dominio algún Nombre Funcional, separado del anterior por el carácter _.

RELACIONES

Propiedades

- Muchas de las propiedades se fundamentan en la Definición Matemática de Conjunto: colección desordenada de elementos distintos del mismo tipo.
- No existen Tuplas repetidas.
En una relación aparece un conjunto de tuplas, y por lo anterior, no pueden haber dos tuplas iguales.
Como consecuencia, siempre existe una Clave Primaria en una relación.
En una tabla si pueden haber filas repetidas.
- Las tuplas no están ordenadas.
De nuevo la definición de conjunto impide el ordenamiento de arriba a abajo.
En una tabla este ordenamiento se define en la representación.
- Los atributos no están ordenados.
Desde que la cabecera se define como un conjunto, esta propiedad se demuestra de modo implícito.
Por la tanto no existe la adyacencia de los atributos, a diferencia de las tablas.
- Los valores de los atributos son atómicos.
Es decir, no existen grupos repetitivos en las relaciones, en los cuales algún dominio está valuado en relaciones.
Esta propiedad es válida para atributos simples y atributos compuestos, que son una concatenación de atributos simples.
En estos casos, la relación está normalizada.

RELACIONES

Ejemplo de Grupo Repetitivo

S#	PC	
	P#	CANT
S1	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
S2	P1	300
	P2	400
S3	P2	200
S4	P2	200
	P4	300
	P5	400

Ejemplo de Normalización

S#	P#	CANT
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

RELACIONES

Tipos de Relaciones

- Relaciones Base o Relaciones Reales.
Es una relación autónoma y con nombre, que tiene la importancia suficiente para que el diseñador la entienda como un elemento específico de su base de datos.
- Vistas o Relaciones Virtuales.
Es una relación derivada que existe mediante su definición en términos de otras relaciones, y no posee datos almacenados.
- Instantáneas.
Relación virtual que sí contiene información, definiéndose a partir de otras relaciones y por los datos que almacenan.
Su creación es muy similar a una consulta, pero además se indica cuando se debe actualizar.
- Resultados de Consultas.
Contiene las tuplas resultantes de una consulta, pero su existencia es efímera.
- Resultados Intermedios.
Relación resultante de alguna expresión relacional dentro de otra expresión relacional mayor.
Un ejemplo es el resultado de una subconsulta.
- Resultados Temporales.
Relación con nombre, similar a una relación base o a una vista, que se destruye de modo automático en algún momento

RELACIONES

Ejemplo de Relación Base en Seudo-SQL

```
CREATE TABLE S
( S#          DOMAIN (S#)          NOT NULL,
  SNOMBRE     DOMAIN (NOMBRE)      NOT NULL,
  SITUACION   DOMAIN (SITUACION)   NOT NULL,
  CIUDAD      DOMAIN (CIUDAD)      NOT NULL,
  PRIMARY KEY (S#) );
```

Ejemplo de Vista en Seudo-SQL

```
CREATE VIEW BUENOS_PROVEEDORES
AS SELECT S#, SITUACION, CIUDAD
FROM S
WHERE SITUACION > 15 ;
```

Ejemplo de Instantánea en Seudo-SQL

```
CREATE SNAPSHOT SC
AS SELECT S#, CIUDAD
FROM S
REFRESH EVERY DAY;
```

Ejemplo de Resultados Intermedio en Seudo-SQL

```
SELECT DISTINCT S.CIUDAD
FROM S
WHERE S.S# IN
( SELECT SP.S#
  FROM SP
  WHERE SP.P# = 'P2' ) ;
```

BASES DE DATOS RELACIONALES

Definición

- Una Base de Datos Relacional es una base de datos percibida por el usuario como una colección de relaciones normalizadas de diversos grados, que varía con el tiempo.

Por "percibida por el usuario" se quiere indicar la visión que el usuario puede observar de la base de datos, que puede diferir del modo real en el que se haya implementado.

El término "varía en el tiempo" indica que el conjunto de tuplas de la relación cambia en el tiempo.

Este último término no se aplica a los dominios subyacentes, ya que en este caso se debería producir un cambio de su definición.

- Una Relación se asemeja a un Fichero "disciplinado", una Tupla a un Registro y un Atributo a un Campo.

Por disciplinado se entiende que:

- El fichero contiene registros de un único tipo.
 - No existe orden en los registros del fichero.
 - No existe orden en los campos del registro.
 - Cada campo tiene un único valor.
 - En cada registro existe una clave primaria.
- Los datos de una base de datos relacional se representan de una y sólo una manera, por su valor explícito.