

BASES DE DATOS IBM AS / 400

D

DEFINICION DE BASE DE DATOS:

Una Base de Datos es un conjunto de datos almacenados en una estructura física y con otra lógica por la cual se relacionan, siendo independiente de las aplicaciones. Tan importante como los datos, es la estructura conceptual con la que se relacionan entre ellos. Un sistema de gestión de bases de datos (DBMS database management system) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la bases de datos. Toda base de datos es una colección de datos tendiente a minimizar la redundancia. Dicha colección de datos permite que los mismos se encuentren

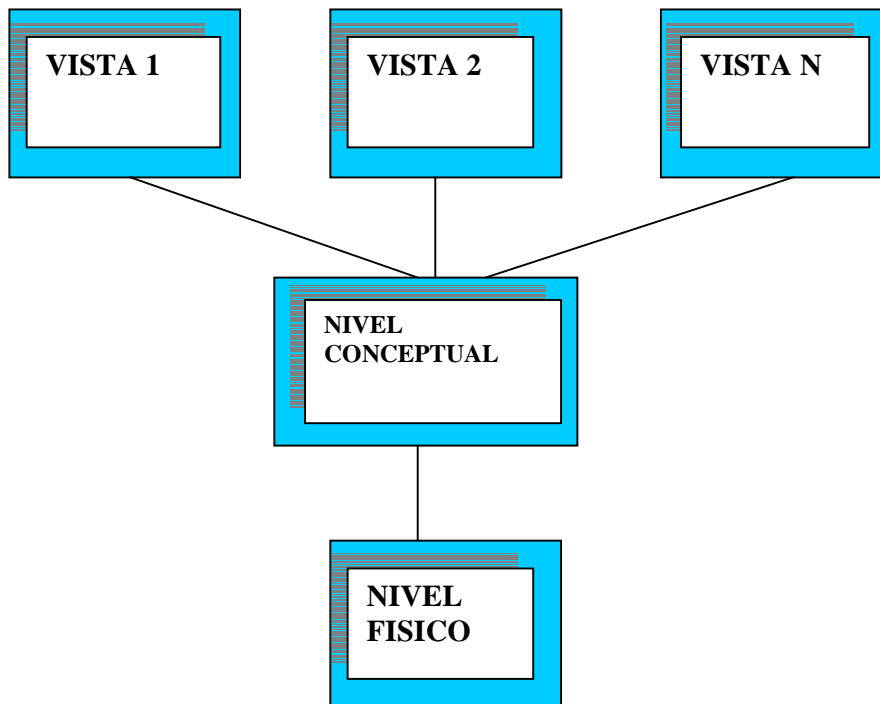
- ❖ Interrelacionados
- ❖ Almacenados en conjuntos
- ❖ Sin redundancias innecesarias o perjudiciales
- ❖ Independientes de los programas que los utilizan

Todo modelo de bases de datos debe gozar de las siguientes características:

- ❖ Independencia de datos
- ❖ Regulación de acceso
- ❖ Protección de integridad
- ❖ Sin redundancia
- ❖ Facilidad de Ordenamiento
- ❖ Manejo centralizado

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores del sistema, independientemente de lo que estos objetos y operadores puedan significar. El principal proceso en el diseño de una base de datos es la creación de un modelo de datos. Este debe representar en escala a la realidad, reflejando la estructura del negocio de la organización por medio de los datos y sus relaciones. En esta abstracción de datos debemos distinguir tres niveles:

- **Nivel físico** : Este es el nivel mas bajo de abstracción y describe como se almacenan los datos
- **Nivel conceptual:** En este nivel se describe que datos son realmente almacenados y las relaciones que existen entre ellos.
- **Nivel de visión:** Este es el nivel más alto y se describe la visión de los usuarios sobre la base de datos.



Al comienzo en la etapa de diseño se genera un modelo conceptual que representa todas las relaciones lógicas y es totalmente independiente de la implementación física. Este es el modelo de entidad – relación. Luego se deberá generar el modelo lógico que no es más que un puente entre el modelo conceptual y el modelo físico. Existen tres tipos de modelo lógico:

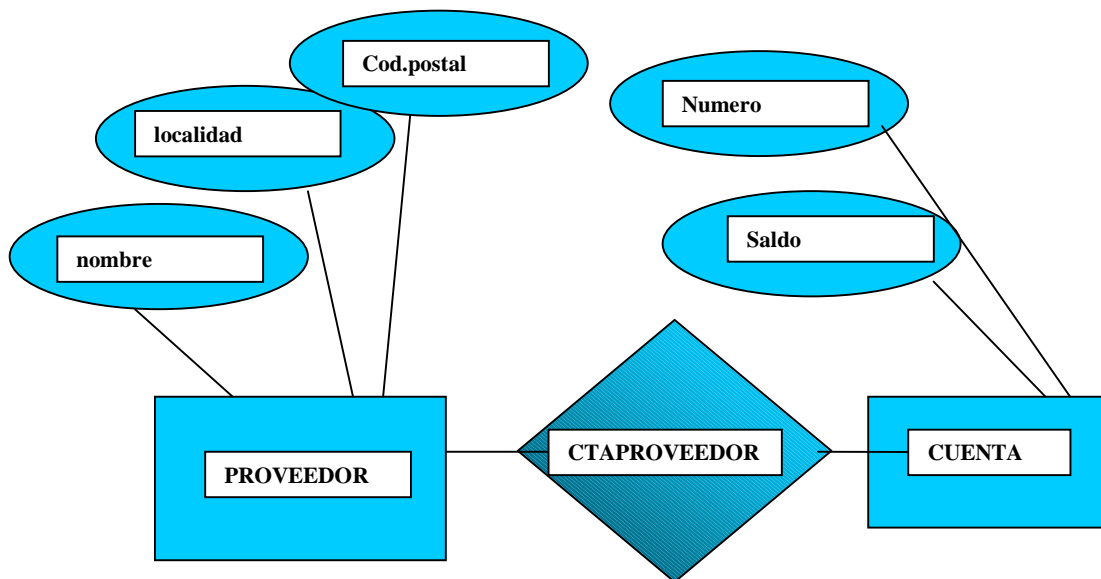
- **Red** : Los datos en este modelo se representan mediante colecciones de registros y las relaciones entre los datos se representan mediante enlaces, los cuales pueden verse como punteros.
- **Jerárquico** : Es similar al modelo anterior en el sentido de que los datos y las relaciones entre datos se representan mediante registros y enlaces.

- **Relacional:** El modelo relacional representa los datos y las relaciones entre los datos mediante una colección de tablas, cada una de las cuales tiene un número de columnas con nombres únicos.

El modelo Relacional se diferencia del modelo de Red y Jerárquico en que no utilizan punteros o enlaces, ya que conecta registros mediante los valores que estos contienen.

Por ultimo nos queda el modelo físico que se construye en base al modelo lógico y describe como serán almacenados los datos, este nivel es el mas bajo.

Particularmente en este manual nos interesa el modelo Entidad-Relación (E – R). Este modelo se basa en una percepción de un mundo real que consiste en una colección de objetos básicos llamados entidades y relaciones entre estos objetos. Una entidad es un objeto que se distingue de otros objetos por medio de sus atributos. Por ejemplo los atributos de la cuenta de proveedores Número y Saldo describen una cuenta en particular. Ahora bien una relación PROVCTA asocia a un proveedor con cada una de las cuentas que tiene. La forma de expresar la estructura lógica global de una base de datos es por medio de un diagrama E-R.



T ERMINOLOGIA DE BASES DE DATOS

Entidad: Es algo que puede ser identificado por sí mismo (personas, lugares, cosas o conceptos) acerca de la cual se requiere guardar información. Un tipo de entidad representa a una clase de entidades que tienen los mismos atributos.

Relación: Es una asociación entre entidades, o sea es la forma en que se asocian las entidades.

Atributo: Es la característica de una entidad

Dato: Son los valores que se le asignan a un atributo de una determinada entidad.

Dominio: Es un conjunto de valores que puede tomar un atributo en una relación.

TIPO DE ENTIDAD	TABLA	RELACION	ARCHIVO
ENTIDAD	FILA	TUPLA	REGISTRO
ATRIBUTO	COLUMNA	ATRIBUTO	CAMPO

Tipo de relaciones:

❖ **Relación uno a uno (1 : 1)**

Una entidad en A esta asociada a lo sumo con una entidad B, y una entidad B esta asociada a lo sumo con una entidad A.

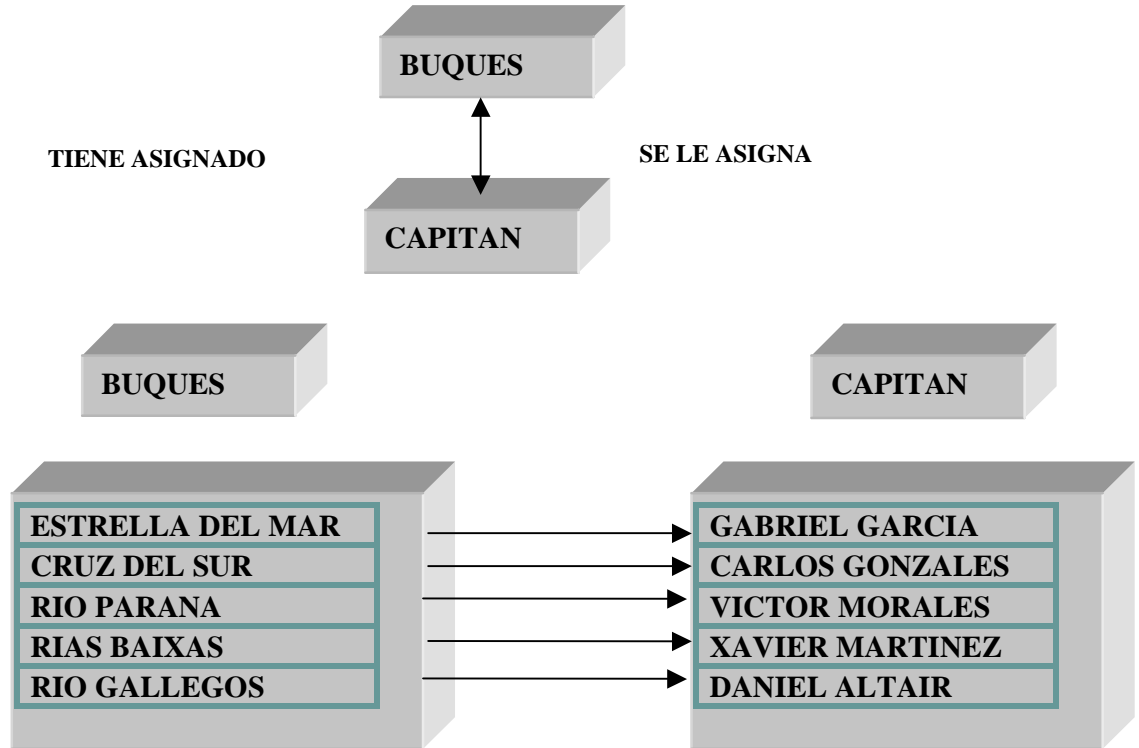
❖ **Relación uno a muchos (1 : M)**

Una entidad en A esta asociada con un numero cualquiera de entidades B. Una entidad B, sin embargo, puede estar asociada a lo sumo con una entidad A.

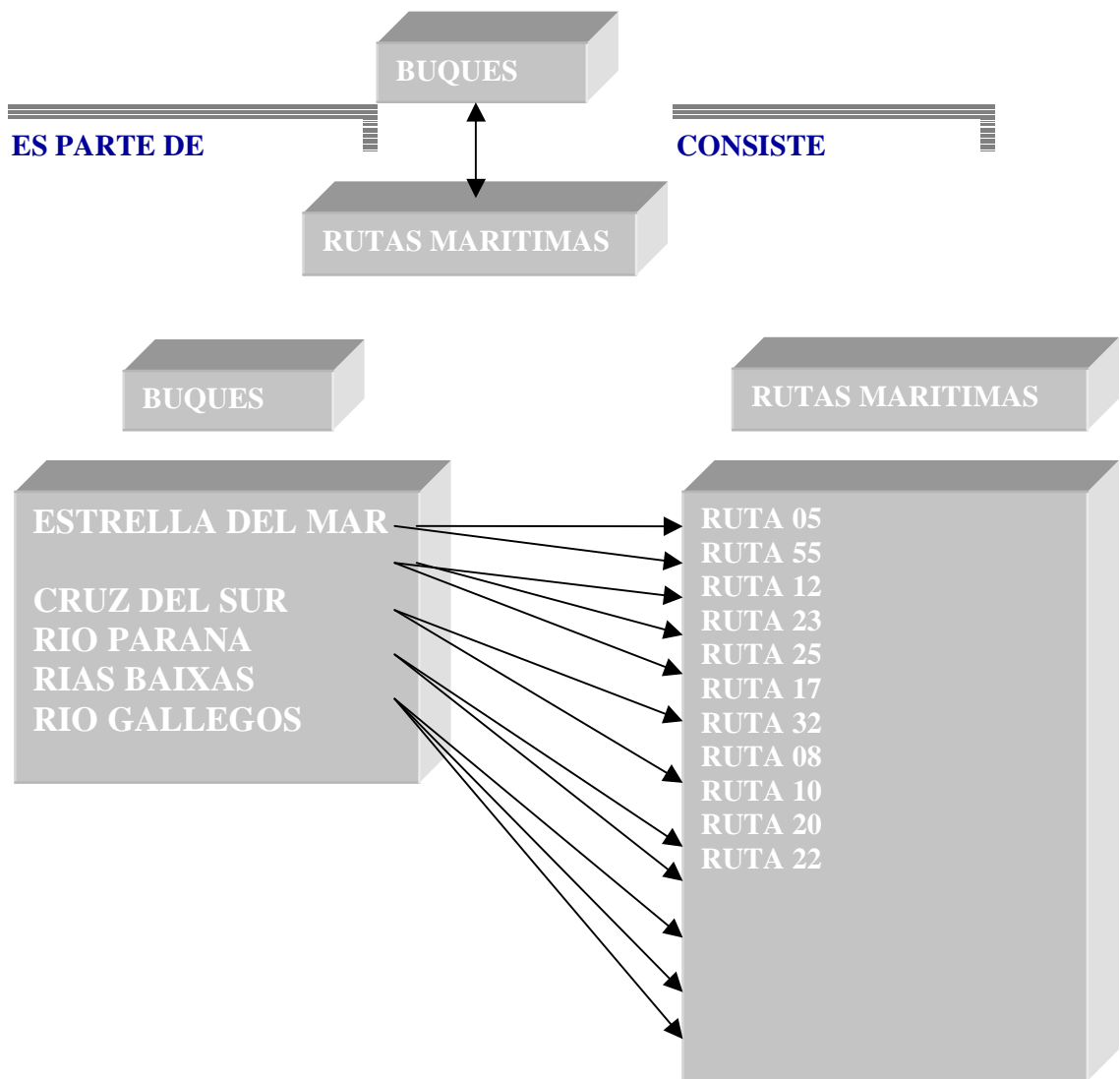
❖ **Relación muchos a muchos (M : M)**

Una entidad en A esta asociada con un numero cualquiera de entidades en B, y una entidad en B esta asociada con un numero cualquiera de entidades en A.

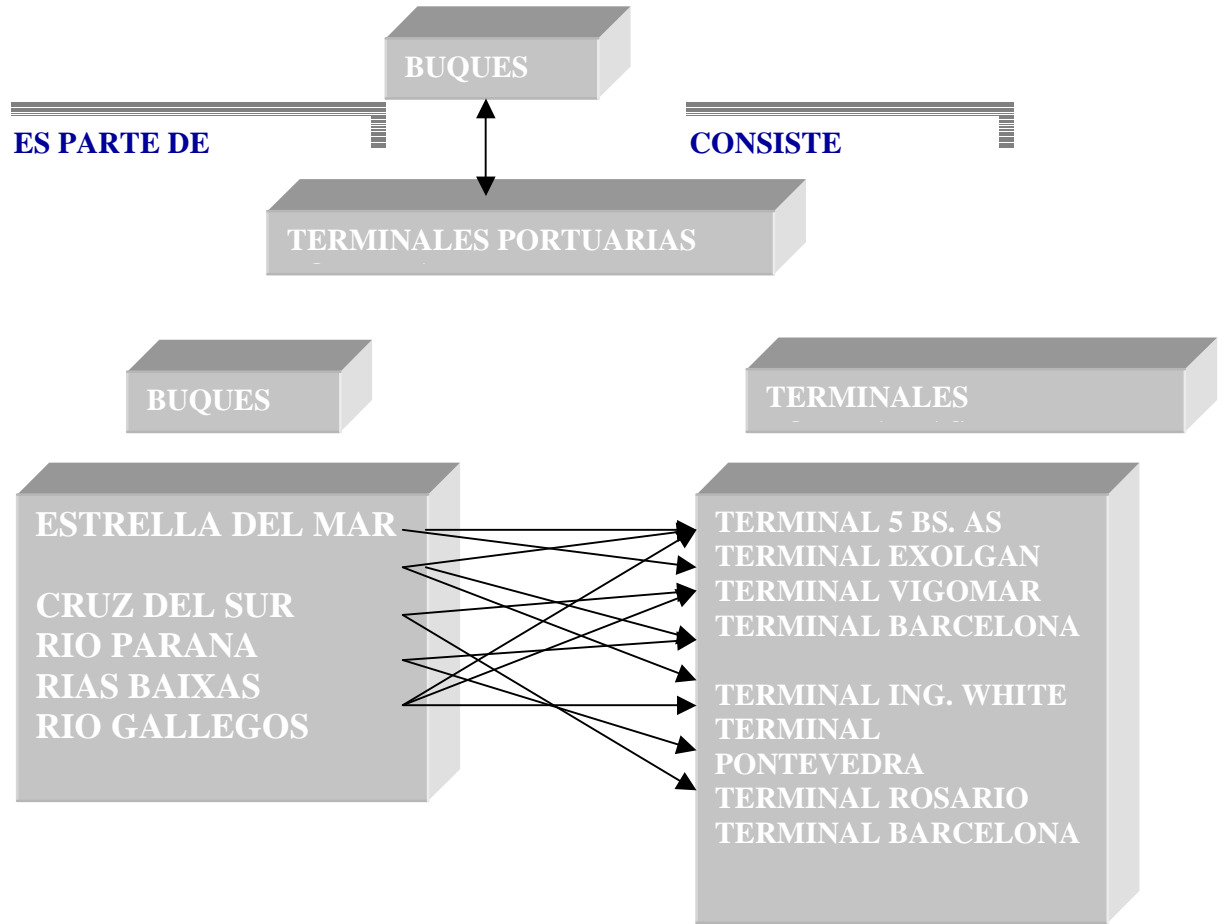
Relación uno a uno (1: 1)



Relación uno a muchos (1: M)



Relación muchos a muchos (M: M)



MODELO DE DATOS

➤ CONCEPTO DE MODELOS DE DATOS

El principal proceso en el diseño de una base de datos es la creación de un modelo de datos. Un modelo de datos es la representación en escala de la realidad. En este modelo reflejaremos la estructura de negocio de la organización, por medio de datos y relaciones.

➤ MODELO CONCEPTUAL

El modelo conceptual deberá reflejar todas las relaciones lógicas y es totalmente independiente de su implementación física. Este es un modelo de entidad relación.

➤ MODELO LOGICO

Este modelo es el puente entre el modelo conceptual y el modelo físico; describe como se verán los datos. Existen en el diseño de bases de datos 3 modelos lógicos:

- 1. Jerárquico**
- 2. Red**
- 3. Relacional**

➤ MODELO FISICO

El modelo físico esta construido sobre las bases del modelo lógico y describe como los datos son almacenados. Este es el nivel mas bajo de abstracción.

NORMALIZACION

Es la técnica utilizada para dividir las estructuras de datos en pequeñas unidades. En estas unidades, cada atributo es totalmente dependiente de la clave primaria de la entidad a la cual pertenece. La normalización nos permite:

- **Minimizar la redundancia**
- **Minimizar el impacto de futuros cambios de datos**
- **Minimiza el mantenimiento de datos**

➤ **DEPENDENCIA FUNCIONAL**

Es la relación existente entre dos atributos, por lo que el conocimiento de uno de ellos determina el valor del otro. Si un elemento A es funcionalmente dependiente de otro elemento B, queda automáticamente definido A. El nombre de un buque es funcionalmente dependiente de su número de matrícula, pero no es cierto que al conocer el nombre del buque se conozca su número de matrícula.

➤ **DEPENDENCIA FUNCIONAL COMPLETA**

Se produce la dependencia funcional completa cuando un atributo tiene dependencia funcional de un conjunto de atributos, pero de ninguno de ellos en particular.

➤ **DEPENDENCIA FUNCIONAL TRANSITIVA**

Se produce la dependencia funcional transitiva cuando un atributo tiene dependencia de otro y este a su vez de un tercero. En este caso, el primero tendrá dependencia transitiva al tercero. Si se tiene los elementos A, B, C, si A es funcionalmente dependiente de B, y B es funcionalmente dependiente de, entonces A es transitivamente dependiente de C.

➤ **PRIMERA FORMA NORMAL**

Una entidad está en primera forma normal si no existen grupos repetitivos.

➤ **SEGUNDA FORMA NORMAL**

Una entidad está en segunda forma normal, si está en primera forma normal, y todos los atributos que no integran la clave están en completa dependencia funcional respecto de la clave.

➤ **TERCERA FORMA NORMAL**

Una entidad esta en tercera forma normal, si esta en segunda forma normal y no existen atributos que no integren la clave que tengan dependencias funcionales respecto de otros atributos no claves.

Veamos un ejemplo en base a bases de datos que podremos observar en distintos ERPs como BPCS y JDE entre otros.

NO NORMALIZADA

ORDEN DE PEDIDO

ORDEN (Numero de Orden, fecha de la Orden, Numero de Cliente, Nombre del Cliente, Estado del cliente, Nro. de Ítem, Descripción, cantidad ordenada, Precio)

Num. Orden	F. de Orden	N. de Cliente	Nombre Cliente	Estado Cliente	Nro. Ítem	Descripción	Cantidad	Precio
0010	20/02/02	91765	CARLOS	A	665	ENGRANAJE 01	20	10,05
0010	20/02/02	91756	CARLOS	A	771	TORNILLO 98	150	1,67
0010	20/02/02	91756	CARLOS	A	668	LLAVE 25 ‘	1	26,98
0555	17/10/02	67943	JOSE	A	771	TORNILLO 98	200	1,67
0555	17/02/02	67943	JOSE	A	665	ENGRANAJE 01	10	10,65

Observemos como se repiten en cada registro dentro de este diseño el número de orden, fecha, numero de cliente, etc.

PRIMERA FORMA NORMAL

ORDEN

Num. Orden	F. de Orden	N. de Cliente	Nombre Cliente	Estado Cliente
0010	20/02/02	91765	CARLOS	A
0555	17/02/02	67943	JOSE	A

LINEA DE ITEM

Num. Orden	Nro. Ítem	Descripción	Cantidad	Precio
0010	665	ENGRANAJE 01	20	10,05
0010	771	TORNILLO 98	150	1,67
0010	668	LLAVE 25 ‘	1	26,98
0555	771	TORNILLO 98	200	1,67
0555	665	ENGRANAJE 01	10	10,65

SEGUNDA FORMA NORMAL

ORDEN

Num. Orden	F. de Orden	N. de Cliente	Nombre Cliente	Estado Cliente
0010	20/02/02	91765	CARLOS	A
0555	17/02/02	67943	JOSE	A

LINEA DE ITEMS

Num. Orden	Nro. Ítem	Cantidad
0010	665	20
0010	771	150
0010	668	1
0555	771	200
0555	665	10

ITEM

Nro. Ítem	Descripción	Precio
665	ENGRANAJE 01	10,05
668	LLAVE 25 ‘	26,98
771	TORNILLO 98	1,67

TERCERA FORMA NORMAL

ORDEN

Num. Orden	F. de Orden
0010	20/02/02
0555	17/02/02

CLIENTE

N. de Cliente	Nombre Cliente	Estado Cliente
91765	CARLOS	A
67943	JOSE	A

LINEA DE ITEM

Num. Orden	Nro. Línea	Nro. Ítem	Cantidad
0010	0001	665	20
0010	0002	771	150
0010	0003	668	1
0555	0001	771	200
0555	0001	665	10

ITEM

Nro. Ítem	Descripción	Precio
665	ENGRANAJE 01	10,05
668	LLAVE 25 ‘	26,98
771	TORNILLO 98	1,67

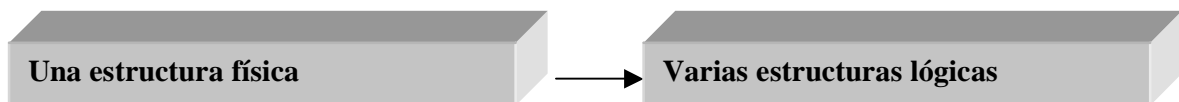
MODELO RELACIONAL

➤ CONCEPTOS SOBRE EL MODELO RELACIONAL

El modelo relacional es el modelo lógico donde las relaciones entre entidades se expresan a través de claves foráneas. Las claves foráneas es la clave primaria de una entidad que se relaciona con un atributo de otra entidad. En el modelo relacional la relación uno a uno se representa en una fila de la tabla. La relación uno a muchos se satisface incorporando en la tabla M claves foráneas., generalmente es la clave primaria de la primer entidad la que se incorpora a la entidad M. La relación muchos a muchos se representan creando una tabla donde la clave primaria estará compuesta por cada una de las claves de las otras relaciones.

➤ BASES DE DATOS RELACIONAL

Una base de datos es relacional cuando esta construida con tablas de ítems de datos. En las mismas las filas caracterizan entidades y las columnas atributos. Estas tablas conforman relaciones entre los datos a partir de las cuales podemos (por medio del álgebra relacional) crear nuevas tablas o relaciones, que se denominan vistas lógicas. El álgebra relacional permite a través de un conjunto de operaciones relacionales sobre las tablas, generar una nueva tabla - resultado.



➤ OPERACIONES RELACIONALES

- **SELECCIÓN:** A partir de una tabla se seleccionan un conjunto de filas, conformando una nueva tabla.

BUQUE	PTO. MATRICULA	TRB	Nombre BUQUE
190010	BS. AS.	47560	RIO PARANA
230555	BS. AS.	32000	CRUZ DEL SUR
125766	VIGO	15000	RIAS BAIXAS
219875	BS. AS.	50000	ESTRELLA DEL SUR



SELECCIÓN
de aquellos buques de más de 40.000 TN de TRB



BUQUE	PTO. MATRICULA	TRB	Nombre BUQUE
190010	BS. AS.	47560	RIO PARANA
219875	BS. AS.	50000	ESTRELLA DEL SUR

- **PROYECCION:** A partir de una tabla se seleccionan un conjunto de columnas, conformando una nueva tabla.

BUQUE	PTO. MATRICULA	TRB	Nombre BUQUE
190010	BS. AS.	47560	RIO PARANA
230555	BS. AS.	32000	CRUZ DEL SUR
125766	VIGO	15000	RIAS BAIXAS
219875	BS. AS.	50000	ESTRELLA DEL SUR



PROYECCION
Buque, nombre



BUQUE	Nombre BUQUE
190010	RIO PARANA
230555	CRUZ DEL SUR
125766	RIAS BAIXAS
219875	ESTRELLA DEL SUR

- **JOIN:** A partir de dos tablas con una o más columnas en común se crea una nueva tabla conteniendo los atributos de las dos tablas iniciales. Tenemos dos tipos de JOIN :

- **INNER JOIN**

- **LEFT OUTER JOIN**

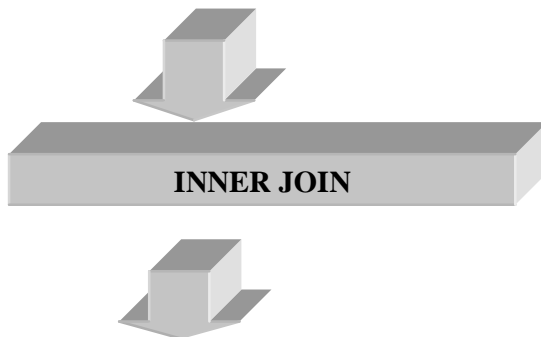
- **INNER JOIN:** Solo las filas que tienen la o las columnas en común sobre la que se hace el join, se incorporan a la tabla que se crea.

ORDEN DETALLES

Num. Orden	N. de Cliente	Nro. Ítem	Cantidad
0010	031184	665	20
0010	031184	771	150
0010	031184	668	1
0555	080191	771	200
0555	080191	665	10

CLIENTE

N. de Cliente	Nombre Cliente	Estado Cliente
91765	CARLOS	A
67943	JOSE	A
200255	DANIEL	A
031184	DIEGO	A
080191	JUAN	A
666666	MARIO	B
717171	GUILLERMO	B



Num. Orden	N. de Cliente	Nro. Ítem	Cantidad	Nombre Cliente	Estado Cliente
0010	031184	665	20	DIEGO	A
0010	031184	771	150	DIEGO	A
0010	031184	668	1	DIEGO	A
0555	080191	771	200	JUAN	A
0555	080191	665	10	JUAN	A

➤ **LEFT OUTER JOIN:** Contiene todas las filas de la primera tabla.

ASIGNATURAS

CODIGO ASIGNATURA	NOMBRE ASIGNATURA
001	MATEMATICA
003	DERECHO MARITIMO
005	CARGA Y ESTIBA
006	CONTABILIDAD
007	MATEMATICA FINANCIERA
008	ADMINISTRACION

ALUMNOS

CODIGO ASIGNATURA	NOMBRE CALUMNO	CALIFICACION
001	CARLOS	8
003	JOSE	7
005	DANIEL	9
006	DIEGO	9
007	JUAN	9
010	MARIO	3
011	GUILLERMO	3

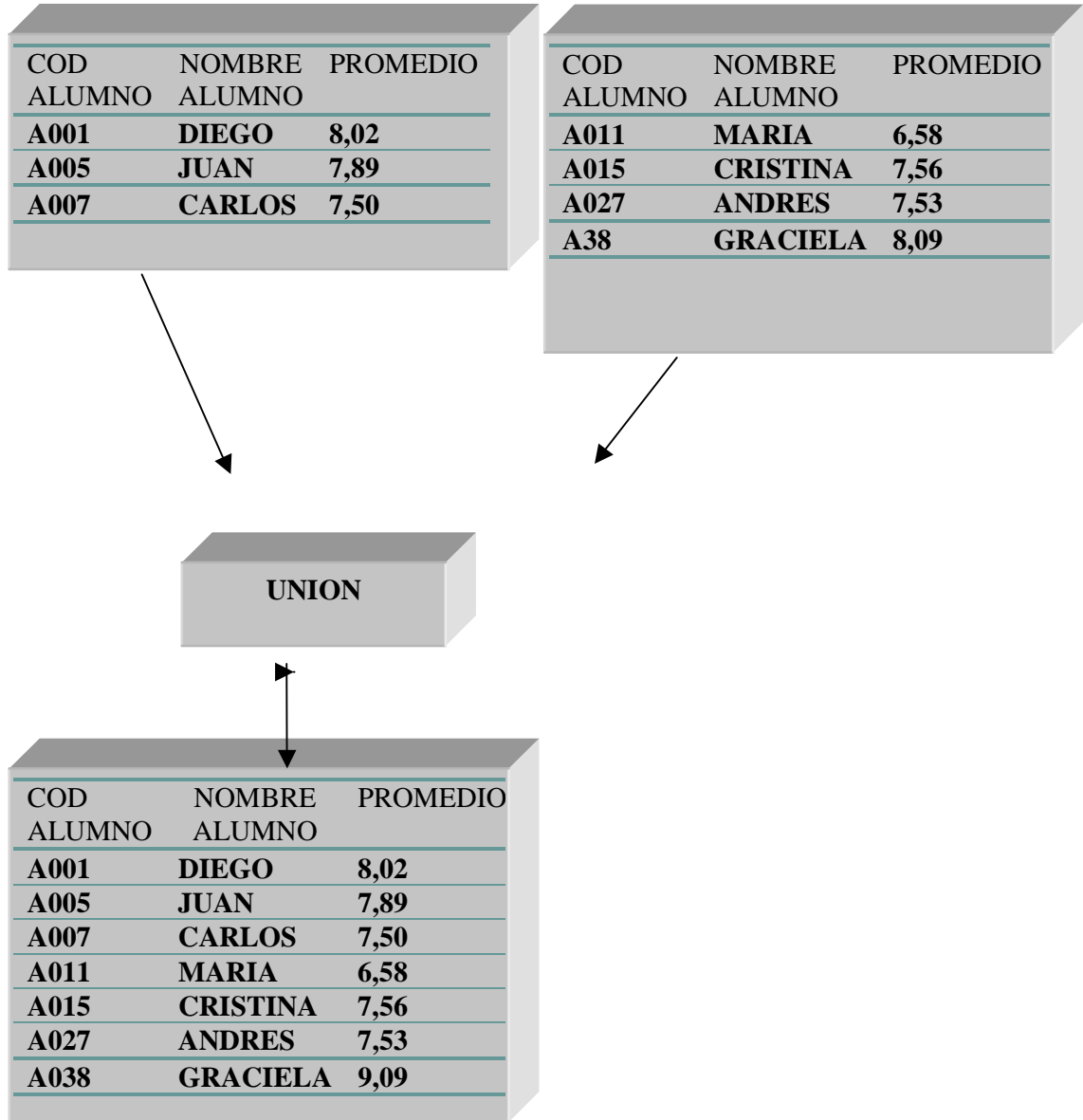


LEFT OUTER JOIN

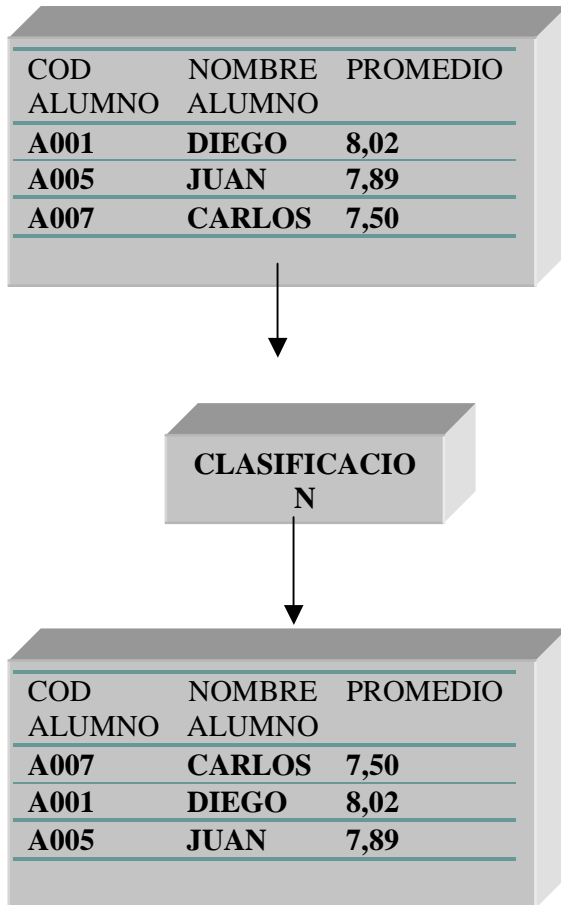


CODIGO ASIGNATURA	ASIGNATURA	NOMBRE CALUMNO	CALIFICACION
001	MATEMATICA	CARLOS	8
003	DERECHO MARITIMO	JOSE	7
005	CARGA Y ESTIBA	DANIEL	9
006	CONTABILIDAD	DIEGO	9
007	MATEMATICA FINANCIERA	JUAN	9
008	ADMINISTRACION		

- **UNION:** A partir de dos tablas se genera una nueva conteniendo las filas de ambas tablas.



- **CLASIFICACION:** A partir de una tabla se genera una nueva tabla reordenada en función de algunos de sus atributos.



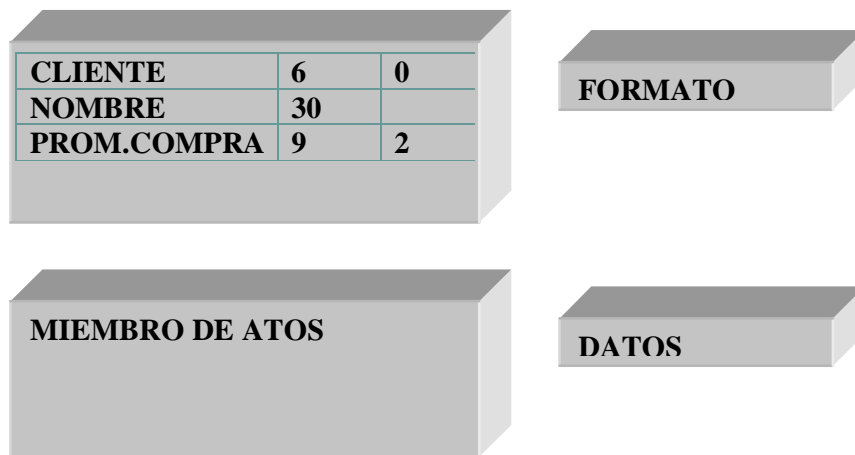
Según lo observado podemos decir que las características de todo modelo relacional son:

- Todos los ítems de datos están en TABLAS BASES.
- Las vistas de los datos en las TABLAS BASES son creadas usando los operadores relacionales de CLASIFICACION, SELECCIÓN, PROYECCION, UNION Y JOIN.
- Cualquier cambio de los datos es reflejado inmediatamente en las vistas.
- No hay punteros dentro de los registros de datos.

❖ EL MODELO RELACIONAL EN EL AS/400:

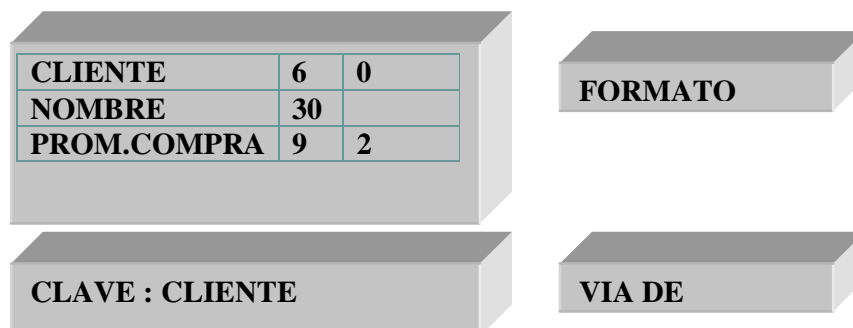
El AS/400 las TABLAS BASES se llaman ARCHIVOS FISICOS. Las VISTAS se llaman ARCHIVOS LOGICOS

▪ ESTRUCTURA DE LOS ARCHIVOS FISICOS



Aunque es posible definir dentro de la estructura de un archivo físico campos clave, es aconsejable que los mismos no contengan ningún campo clave.

▪ ESTRUCTURA DE LOS ARCHIVOS LOGICOS



CODIFICACION DE ARCHIVOS FISICOS

A continuación explicaremos la codificación de DDS – DATA DESCRIPTION SPECIFICATIONS para los archivos físicos.

Define el registro

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	BUQUES					
A		CODBUQ		6		0	
A		NOMBUQ		30			
A		ESLORA		9		2	
A		MANGA		9		2	
A		PUNTAL		9		2	
A		TRB		9		2	
A		CALADO		9		2	
A	K	CODBUQ					

R = Define el nombre de registro

K= Define cual o cuales son los campos clave/s

En la columna longitud se establece la longitud del campo., y en la columna posiciones decimales se coloca la cantidad de decimales que tendrá el campo numérico.

Tipos de datos: En esta columna se pueden determinar los siguientes tipos de datos

- P - Decimal empaquetado
- S – Decimal por zona
- B – Binario
- F – Punto flotante
- A – Carácter

Si no colocamos en esta columna ningún valor asume por omisión los siguientes valores

1. Si el campo definido es numérico o sea en la columna posiciones decimales tiene un valor entre 0 y 31 por default define el tipo de datos como P (decimal por zona).
2. Si el campo es de caracteres (alfanumérico) o sea en la columna posiciones decimales esta en blanco, asume por default A

❖ **DEFINICION DE PALABRAS CLAVES**

Hay cuatro niveles de definición de palabras claves:

- A nivel de archivo
- A nivel de registro
- A nivel de campo
- A nivel de clave

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
							Nivel archivo
A	R	BUQUES					Nivel registro
A		CODBUQ		6		0	
A		NOMBUQ		30			
A		ESLORA		9		2	
A		MANGA		9		2	Nivel campos
A		PUNTAL		9		2	
A		TRB		9		2	
A		CALADO		9		2	
A	K	CODBUQ					Nivel clave

▪ **PALABRAS CLAVES A NIVEL DE ARCHIVO**

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
							REF(BIB01/GJH)
A	R	BUQUES					
A		CODBUQ		6		0	
A		NOMBUQ	R				

A		ESLORA	R		
A		MANGA	R		
A		PUNTAL	R		
A		TRB		9	2
A		CALADO		9	2
A	K	CODBUQ			

REF a nivel de archivo especifica en que biblioteca y archivo (GJH) se encuentran los formatos de campos que tienen una R en la columna Ref. Además de esta palabra clave existen otras como FIFO y LIFO que nos indican la forma en que se recuperaran los datos, utilizándose esta palabra solo en el caso que exista claves duplicadas. Otra palabra es UNIQUE que indica que no pueden existir claves duplicadas.

- PALABRAS CLAVES A NIVEL DE REGISTRO

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	BUQUES					FORMAT(GJH) TEXT('ARCHIVO DE BUQUES')
A		CODBUQ		6		0	
A		NOMBUQ		30			
A		ESLORA		9		2	
A		MANGA		9		2	
A		PUNTAL		9		2	
A		TRB		9		2	
A		CALADO		9		2	

La palabra FORMAT indica que el formato de registro será el mismo del archivo GJH, y la palabra TEXT nos permite asociar un comentario de hasta 50 caracteres al formato de registro.

- PALABRAS CLAVES A NIVEL DE CAMPO

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	BUQUES					FORMAT(GJH) TEXT('ARCHIVO

A		CODBUQ		6		0	DE BUQUES') ALIAS(SHIP)
A		NOMBUQ		30			
A		ESLORA		9	F	2	FLTPCN(*SINGLE)
A		MANGA		9		2	
A		PUNTAL		9		2	
A		TRB		9		2	TEXT('TONELADA DE REGISTRO BRUTO')
A		CALADO		9		2	DFT('32')

La palabra ALIAS nos permite definir un nombre alternativo para el campo de hasta 30 caracteres. FLTPCN especifica la precisión de los campos de punto flotante, * SINGLE o * DOUBLE. DFT nos permite definir valores por default del campo y TEXT nos permite introducir un comentario de hasta 50 caracteres. Además de las palabras claves a nivel de campo utilizadas en este ejemplo se cuenta con otras a saber:

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	BUQUES					FORMAT(GJH) TEXT('ARCHIVO DE BUQUES')
A		CODBUQ		6		0	
A		NOMBUQ	R				REFFLD(BIB01/GJH)
A		ESLORA	R				
A		MANGA	R	+2			REFFLD(ESLORA *SRC)
A		PUNTAL		9		2	
A		TRB		9		2	
A		CALADO		9		2	

La primera palabra clave REFFLD indica que el formato del campo se encuentra en otro archivo de referencia distinto al nombrado en la palabra clave a nivel de archivo. La segunda palabra REFFLD nos indica que el formato del campo es el mismo que el campo ESLORA mas dos posiciones.

- PALABRAS CLAVES A NIVEL DE CLAVE

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
---------------	----------------	--------	------	-------	---------------	-----------	-----------

A	R	BUQUES			FORMAT (GJH) TEXT('ARCHIVO DE Buques')
A		CODBUQ		6	0
A		NOMBUQ	R		
A		ESLORA	R		
A		MANGA	R	+2	
					REFFLD(BIB01/GJH)
					REFFLD(ESLORA *SRC)
A		PUNTAL		9	2
A		TRB		9	2
A		CALADO		9	2
A		CODBUQ			ABSVAL

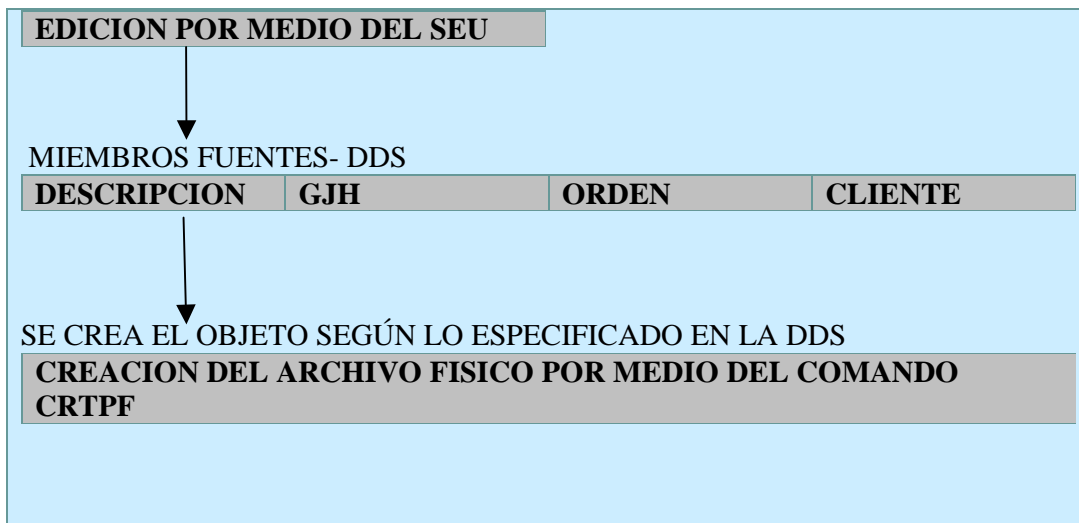
La palabra clave **ABSVAL** nos indica que se ordenara por el valor absoluto del campo. Se dispone de otras palabras claves como:

UNSIGNED: Los registros se ordenaran según los valores hexadecimales, es decir, de acuerdo a su posición en el código 'EBCDIC'.

SIGNED: Los registros se ordenaran según su valor numérico.

DESCEND: Los registros se ordenaran según el orden descendente de la clave.

- **CREACION DE UN ARCHIVO FISICO**



COMANDO CRTPF:

FILE (*CURLIB/MONBRE-ARCHIVO)
SRCFILE(*LIBL/QDDSSRC)

**SRCMBR(*FILE)
MBR(*FILE)
MAXMBRS(1)**

Si necesitamos por ejemplo 3 miembros en un archivo el comando seria

**FILE (*CURLIB/MONBRE-ARCHIVO)
SRCFILE(*LIBL/QDDSSRC)
SRCMBR(*FILE)
MBR(*FILE)
MAXMBRS(3)**

En muchos casos por sobre todo en ERPs como BPCS y JDE es aconsejable que la creación sea con N miembros

**FILE (*CURLIB/MONBRE-ARCHIVO)
SRCFILE(*LIBL/QDDSSRC)
SRCMBR(*FILE)
MBR(*FILE)
MAXMBRS(*MAX)**

▪ **COPIAR UN ARCHIVO FISICO:**

Para copiar un archivo físico utilizaremos el comando CPYF

COMANDO CPYF

CPYF FROMFILE ()

TO FILE ()

MBROPT { ***REPLACE**
 ***ADD**

FMTOPT { ***MAP**
 ***DROP**

CRT FILE { ***NO**
 ***YES**

Si deseamos visualizar los datos de un archive sin utilizar SQL o Query podemos utilizar el comando DSPPFM

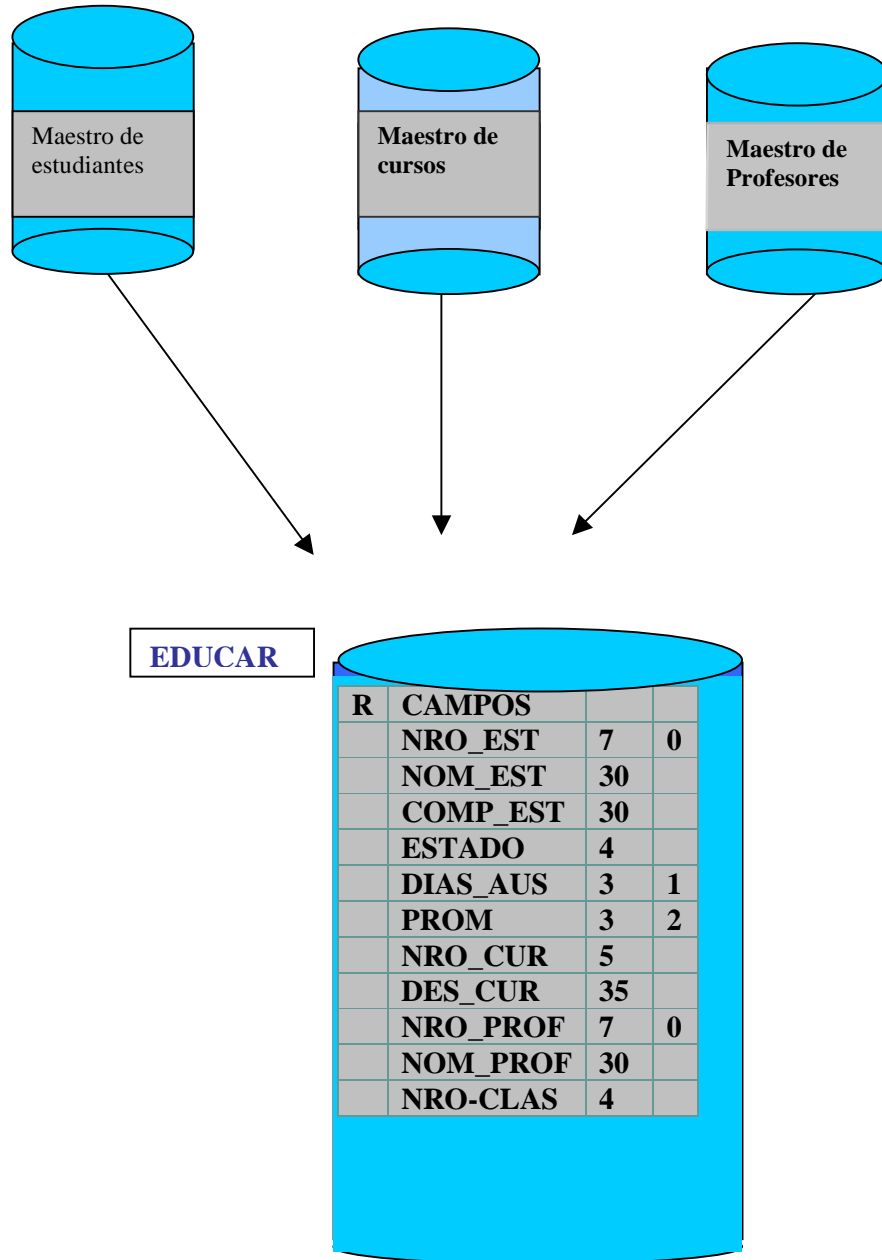
COMANDO DSPPFM

DSPPFM FILE (NOMBRE DE LA BIBLIOTECA / NOMBRE DEL ARCHIVO) MBR (*FIRST)



ARCHIVOS DE REFERENCIA DE CAMPOS

Se justifica el uso de los archivos de referencia de campo cuando se tienen campos de la Base de datos presentes en múltiples registros.



Luego la codificación de la DDS de un archivo se hace en base al archivo de referencia, por ejemplo:

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. de Dec.	Funciones
							REF(BIB01/EDUCAR)
A	R	PROFE					
A		NRO_PROF	R				
A		NOM_PROF	R				
A		NRO_CLAS	R				

❖ FORMAS DE ESTRUCTURAR UN ARCHIVO DE REFERENCIA DE CAMPOS

- En secuencia alfabética
- Por registro
- En secuencia alfabética y luego por formato de registro

El uso de archivos de referencia de campos trae varias ventajas a saber:

- Reduce el esfuerzo de codificación de DDS
- Se asegura la consistencia en el uso de los atributos de archivos.
- Se simplifica cualquier cambio requerido.



ARCHIVOS LOGICOS

❖ ARCHIVOS LOGICOS ‘NONJOIN ‘

En los archivos lógicos también existen las palabras claves en sus niveles de ARCHIVO, REGISTRO, CAMPO, CLASE. Y SELECT / OMIT.

- Palabras Claves a nivel de Archivo

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
							ACCPT(BIB01/GJH)
A	R	PROFE					
A		NRO_PROF					
A		NOM_PROF					
A		NRO_CLAS					

La palabra clave ACCPT permite compartir un vía de acceso, no debiéndose de definir los campos claves ni de select / omit.

Otras palabras claves que podemos utilizar a nivel de archivos lógicos son:

FIFO o LIFO: Los registros con claves duplicadas son accedidos en ese orden. **FIRST – IN – FIRST – OUT o LAST – IN –FIRST – OUT.**

UNIQUE: No acepta claves duplicadas.

DYNSLT: Selecciona los registros cuando se abra el archive.

- Palabras Claves a nivel de registro.

- Archivo lógico NON – JOIN sobre un único archivo físico

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	PROF01					PFILE(BIB01/PROF)
A		NRO_PROF					

A	NRO_EST
A	NOM_EST

Con la palabra clave PFILE se direcciona al archivo físico sobre el cual se va a operar. No se especifican campos pues se asume que se utilizaran todos. En el ejemplo que veremos a continuación solo tendrá los campos nombrados.

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	PROF01					PFILE(BIB01/PROF)
A		NRO_PROF					
A		NOM-PROF					
A	K	NRO_PROF					

- Archivo lógico NON – JOIN sobre múltiples archivos físicos y con un único formato.

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	REG01					PFILE(PROF ALUM)
A		NRO_PROF					
A		NOM-PROF					
A		NRO_EST					
A		NOM_EST					
A	K	NRO_PROF					
A	K	NRO_EST					

El archivo lógico resultante tendrá todos los campos nombrados de ambos archivos.

- Palabras Claves a nivel de campo clave

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	REG01					
A		NRO_PROF					
A		NOM-PROF					
A	K	NRO_PROF					ABSVAL

La palabra ABSVAL ordena por valor absoluto. Además tenemos otras palabras como ser:

UNSIGNED: Considera al campo clave no como si fuera un campo numérico sino como si fuera de caracteres.

SIGNED: Es el valor que se toma por omisión para los campos numérico.

DESCEND: Es valido para los campos de carácter y numéricos permitiéndonos obtener una vista en orden descendiente.

DIGIT: Se ignora la posición de zona de cada BIT. Las mismas son consideradas como si estuvieran completas con ceros binarios.

ZONE: Es lo opuesto a la palabra clave anterior.

- Palabras Claves a nivel de SELECT / OMIT

Las palabras claves SELECT y OMIT controlan que registros aparecen en el archivo lógico. Las mismas se especifican al final del archivo lógico. S = SELECT y O = OMIT.

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	REGIS					
A		NROBUQ					
A		NOMBRE					
A		ESLORA					
A		MANGA					
A		TRB					
A		PTOMAT					
A	S	NROBUQ					COMP (GE 200)
A							COMP (LE

A	S	ESLORA	1000)
			RANGE (1000
			200)
A	S	TRB	COMP (LT
			45000)

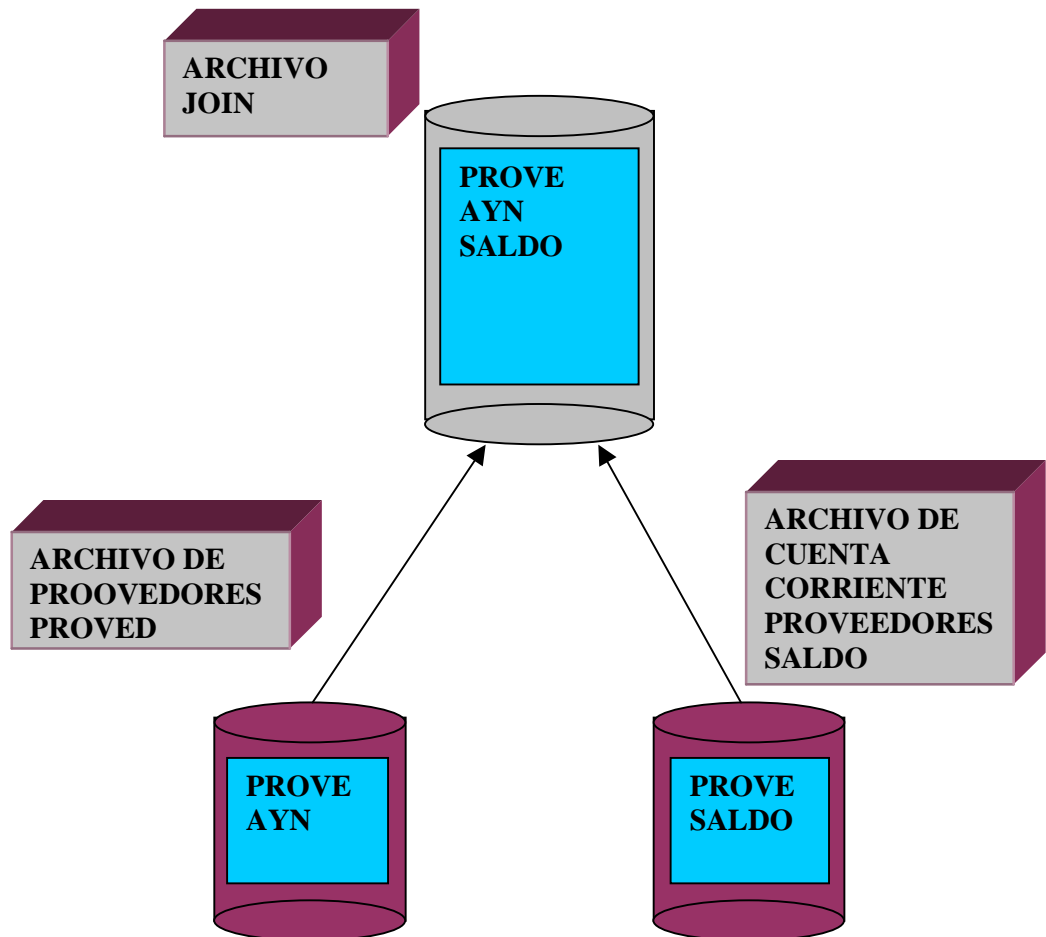
Con una sola S o una O en una línea y un blanco en la siguiente, se hace un AND de las líneas. Con una S o una O en varias líneas se realiza un OR

Para crear un archivo lógico seguiremos el siguiente paso:

1. Se crea la DDS por medio del SEU
2. Se crea el archivo lógico según lo especificado en la DDS utilizando el comando CRTLF.

FILE (*CURLIB / NOMBRE – ARCHIVO)
SRCFILE (*LIBL/ QDDSSRC)
MBR (*FILE)
MAXMBRS (1)
DTAMBR (* ALL)
TEXT (*SRCMBRTXT)

❖ ARCHIVOS LOGICOS ‘ JOIN ‘



El los archivos JOIN se combinan en un formato, campos de dos o más archivos físicos. Solo un formato de registro esta permitido, los campos claves deben ser del archivo considerado primario y estos archivos son de solo lectura. Si realizamos un JOIN de archivos secundarios solo lo podremos realizar en el orden de la palabra clave JFILE.

Tipo de Form.	Tipo de nombre	Nombre	Ref.	Long.	Tipo de Datos	Pos. Dec.	Funciones
A	R	REJOIN					JFILE(PROVE + SALDO)
A	J						JOIN(PROVE + SALDO)
A							JFLD(PROVE PROVE)
A		PROVE					JREF(PROVED)
A		AYN					
A		SALDO					
A	K	PROVE					

AS/400 EN EL MANEJO DE BASE DE DATOS

Los archivos de bases de datos en el AS /400 pueden ser definidos en los lenguajes RPG/400, COBOL/400 , etc, de dos formas diferentes:

- En forma externa
 - Dentro de un programa RPG, COBOL, PL/I, etc.
-
- **Ejemplo de un archivo descrito externamente en un programa COBOL/400**

ENVIROMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE CONTROL.
SELECT MAECUR.
ASSIGN TO DATABASE – CURSO.
ORGANIZATION INDEXED.
ACCES IS RANDOM.
RECORD KEY IS NROCUR.
DATA DIVISION.
FILE SECTION.
FD MAECUR.
LABEL RECORD ARE STANDARD.
01 REGMAE
COPY DDS – REGCUR OF CURSO.

- **Ejemplos de un archivo descrito externamente en un programa RPG/400.**

CURSO U F E K DISK

CURSO O F E K DISK A