

APÉNDICE D. INTRODUCCIÓN A SQL

D.1 INTRODUCCIÓN. CONCEPTOS PREVIOS

D.1.1 Base de Datos Relacional

Conjunto de Datos que el usuario percibe como una colección de tablas. La visión tabular de los datos es independiente de cómo se almacenan.

D.1.2 Tabla Relacional

Cumple las siguientes características:

- Es un conjunto de Filas y Columnas.
- Cada Fila representa un registro, y es un grupo de columnas que da toda la información necesaria sobre un sujeto determinado de la tabla.
- Cada columna tiene un nombre que identifica toda la información que hay en ella.
- El orden de las filas y columnas no es significativo.

D.1.3 El Lenguaje

Dotado de las siguientes características:

- Orientado hacia el usuario final.
- De alto nivel.
- Puede utilizarse de forma interactiva o por programa.
- Para definición de datos (DDL), y manipulación de Datos (DML). Este último realiza funciones de:
 - Recuperación: Selección, Proyección, Join, Unión.
 - Modificación.
 - Actualización, Borrado, Inserción.

Toda operación de recuperación o modificación sobre una tabla, da como resultado otra tabla.

En los ejemplos que siguen utilizaremos las siguientes tablas:

Tabla de Pólizas: POLIZA

NUMPOLI	NUMCLIE	TIPOPOLI
1001	A001	INDIVIDUAL
1002	A002	INDIVIDUAL
2001	A003	COLECTIVA

Tabla de Clientes: CLIENTE

NUMCLIEN	NOMBCLIE
A001	PEREZ
A002	SANCHEZ
A003	GARCIA
A004	MARTINEZ

D.1.3.1 Selección

La operación de Selección recupera un conjunto de filas de una tabla, cumpliendo además una cierta condición, si se desea.

Ejemplo: "Seleccionar todas las pólizas individuales"

NUMPOLI	NUMCLIE	TIPOPOLI
1001	A001	INDIVIDUAL
1002	A002	INDIVIDUAL

D.1.3.2 Proyección

La operación de proyección recupera un grupo determinado de columnas de una tabla.

Ejemplo: "Recuperar los números de Cliente y Póliza"

NUMPOLI	NUMCLIE
1001	A001
1002	A002
2001	A003

D.1.3.3 Join

Combinación de las filas de una tabla con filas de otra tabla, basándonos en los valores de una determinada columna.

Ejemplo: "Recuperar Números de Póliza y Nombres de los Titulares"

NUMPOLI	NOMBCLIE	TIPOPOLI
1001	PEREZ	INDIVIDUAL
1002	SANCHEZ	INDIVIDUAL
2001	GARCIA	COLECTIVA

D.2 SQL. CONCEPTOS BÁSICOS

SQL (*Structured Query Language*) nace como lenguaje de DB2: Base de Datos relacional para los grandes sistemas de IBM.

Con SQL, el usuario puede extraer los datos especificando *qué* quiere, y no *cómo* lo quiere. El usuario no tiene por qué conocer la disposición de los datos en el almacenamiento físico.

Con una sentencia SQL se puede acceder a:

- Una o más columnas de una fila de una tabla.
- Una o más columnas de múltiples filas de una tabla.
- Una o más columnas de múltiples filas de múltiples tablas.
- Una o más columnas calculadas a partir de otras columnas.

D.2.1 Select Básica

Es la operación más sencilla y utilizada de SQL. Permite extraer información de una base de datos relacional.

Ejemplo: "Recuperar de la tabla de pólizas la póliza número "1001"

```
SELECT  NUMPOLI, NUMCLIE, TIPOPOLI
FROM    POLIZA
WHERE   NUMPOLI = '1001'
```

NUMPOLI	NUMCLIE	TIPOPOLI
1001	A001	INDIVIDUAL

SELECT es una de las 4 sentencias básicas del lenguaje de manipulación (DML):

- SELECT: recupera datos.
- UPDATE: actualiza.
- DELETE: borra una o más filas.
- INSERT: inserta filas

SELECT recupera un grupo específico de registros Su sintaxis es:

SELECT < lista-campos | * > FROM nomb-tabla WHERE condición

Donde:

- ‘*’ indica que se desean todas las columnas en el mismo orden con el que se definieron en la tabla.
- ‘FROM’ especifica la tabla de búsqueda.
- ‘WHERE’ puede incorporar condiciones a las filas deseadas.
- ‘lista-campos’ especifica sólo ciertas columnas de la tabla.

D.2.2 Concepto de Join

Es una ‘Select’ desde varias tablas. Es posible siempre que las tablas del Join tengan algún dato en común.

La operación de JOIN relaciona dos o más tablas, permitiendo de esta forma que datos de varias tablas se accedan juntos para obtener un resultado que no estaría disponible desde una única tabla.

Ejemplo: Listar el número de póliza, número de cliente y nombre de los titulares de las pólizas. Por cada fila en POLIZA, encontrar la fila en CLIENTE, que contenga en la columna NUMCLIEN un valor igual al de la columna NUMCLIE de la primera tabla. De estas filas poner los valores de las columnas seleccionadas (SELECT), en la tabla resultado.

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE
FROM    POLIZA, CLIENTE
WHERE   NUMCLIE = NUMCLIEN
```

NUMPOLI	NUMCLIE	NOMBCLIE
1001	A001	PEREZ
1002	A002	SANCHEZ
2001	A003	GARCIA

D.2.3 Actualizar Datos (UPDATE)

UPDATE tabla
 SET nuevos valores en los campos
 WHERE condiciones de actualización

Ejemplo:

UPDATE POLIZA
 SET NUMPOLI = '2002'
 WHERE NUMPOLI = '2001'

NUMPOLI	NUMCLIE	TIPOPOLI
1001	A001	INDIVIDUAL
1002	A002	INDIVIDUAL
2002	A003	COLECTIVA

D.2.4 Insertar Datos (INSERT)

Tiene dos formas básicas, según queramos insertar una o varias filas:

1.-INSERT INTO tabla
 (columnas)
 VALUES (lista de valores)

Ejemplo:

INSERT INTO POLIZA
 (Numpoli,
 Numclie,
 Tipopoli)
 VALUES
 ('2002','A004', 'colectiva')

NUMPOLI	NUMCLIE	TIPOPOLI
1001	A001	INDIVIDUAL
1002	A002	INDIVIDUAL
2001	A003	COLECTIVA
2002	A004	COLECTIVA

Si no se especifican las columnas se asume que los valores se insertan en el orden en que se definieron las columnas.

2.-INSERT INTO tabla
 (columnas)
 SELECT

Cada fila recuperada con SELECT se insertará en la tabla.

D.3 SQL AVANZADO

Utilizaremos la siguiente tabla en los ejemplos:

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

D.3.1 CRITERIOS DE BUSQUEDA (ORDER BY)

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   TIPOPOLI = Individual
ORDER BY NOMBCLIE
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
2001	A003	GARCIA	010191	010192	2000	Individual
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual

La secuencia en que se obtienen las columnas en una SELECT viene determinada por su posición en la cláusula SELECT. La secuencia de las filas viene determinada en principio por el orden físico en que se encuentran almacenadas.

Si el usuario quiere ordenar la salida debe utilizar la cláusula ORDER BY, seguida de los nombres de las columnas que controlan el orden en que desea que aparezcan las filas, teniendo en cuenta que la operación por defecto en la ordenación es ascendente. Si se quiere el orden descendente, hay que codificar.

ORDER BY nombre-columna DESC

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
ORDER BY NUMPOLI DESC
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
2002	B004	MARTINEZ	010192	010192	2500	Colectiva
2001	A003	GARCIA	010191	010192	2000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
1001	A001	PEREZ	010191	010192	1000	Individual

D.3.2 OPERADORES

SQL soporta los siguientes operadores de comparación:

=	Igual a
<>	No igual a
>	Mayor que
>=	Mayor o Igual a
<	Menor que
<=	Menor o igual que
<><	No menor que
<>>	No mayor que

Los operadores de comparación pueden usarse para comparar el valor de una columna con un valor determinado, o comparar el valor de dos columnas.

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   RECITOTA = 1000
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA,
        FECHBAJA, RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   FECHBAJA > FECHALTA
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual

D.3.3 EXPRESIONES

SQL soporta las siguientes expresiones:

+	Suma
-	Resta
*	Multiplicación
/	División

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   FECHBAJA - FECHALTA = 0
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

D.3.4 IN y BETWEEN

Permiten realizar una selección desde una lista o rango de valores.

WHERE nombre-col IN (value, value,...)

WHERE nombre-col BETWEEN value AND value

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NUMCLIE IN ('A001','A002')
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NUMCLIE BETWEEN 'A001' AND 'A003'
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual

D.3.5 NOT

Cualquiera de las cláusulas vistas anteriormente puede ir precedidas con NOT, como opuesto a la condición especificada.

WHERE nombre-col NOT IN (value, value,...)

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NUMCLIE NOT IN ('A001','A002','A003')
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

D.3.6 LIKE

Permite la búsqueda de caracteres de datos sin necesidad de especificar todos ellos. Su sintaxis es:

WHERE nombre-col LIKE (ccc <%><_>)

Donde:

- ‘ccc’ es un caracter o string de caracteres.
- ‘%’ representa un conjunto de cero o más caracteres.
- ‘_’ representa un único carácter.

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NOMBCLIE LIKE 'B%'
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NOMBCLIE LIKE '_A%'
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

D.3.7 AND y OR

Permiten combinar condiciones de búsqueda.

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NOMBCLIE LIKE '_A%' AND RECITOTA = 1500
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1002	A002	SANCHEZ	010191	010192	1500	Individual

Ejemplo:

```
SELECT  NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
        RECITOTA, TIPOPOLI
FROM    POLIZA
WHERE   NOMBCLIE LIKE '_A%' AND (RECITOTA = 1500 OR RECITOTA =
        2000)
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual

D.3.8 JOIN

Operación que realiza búsquedas en varias tablas a la vez.

Ejemplo:

```
SELECT  NUMPOLI, NOMBCLIE
FROM    POLIZA, CLIENTE
```

NUMPOLI	NOMBCLIE
1001	PEREZ
1002	SANCHEZ
2001	GARCIA

En la cláusula FROM, deben especificarse las dos tablas implicadas. El JOIN puede hacerse sobre más de dos tablas. Si las dos tablas tuvieran una columna con el mismo nombre, hay que calificar el nombre con:

NOMBRE DE TABLA.NOMBRE DE COLUMNA

O bien utilizar otro tipo de calificador. En este caso, se procede como sigue:

- Se elige el calificador en la cláusula FROM (en el ejemplo siguiente, P es el calificador de POLIZA y C el de CLIENTE).
- Se utiliza el nuevo calificador para indicar a que tabla pertenece una columna determinada.

Ejemplo:

```
FROM    POLIZA P, CLIENTE C
WHERE   P.NUMCLIE = C.NUMCLIE
```

D.3.9 DISTINCT

Elimina la obtención de filas duplicadas que puedan existir en una tabla.

Para la tabla:

NUMPOLI	NUMCLIE	NOMBCLIE
1	1	PEREZ
2	1	PEREZ
3	2	SANCHEZ

Ejemplo:

```
SELECT  DISTINCT NUMCLIE, NOMBCLIE
FROM    POLIZA
```

NUMPOLI	NUMCLIE	NOMBCLIE
1	1	PEREZ
3	2	SANCHEZ

EXPRESIONES

Son las mismas operaciones vistas anteriormente, pero pueden ir no sólo en la cláusula WHERE, sino en la propia SELECT.

De esta forma le podemos pedir al SQL que nos cree una nueva columna, y además podemos utilizarla referenciándola por su posición en la SELECT en un ORDER BY.

Ejemplo:

```
SELECT  NUMCLIE, NOMBCLIE, RECITOTA/12
FROM    POLIZA
WHERE   NUMCLIE LIKE 'A%'
ORDER BY 3      (Hace referencia a la tercera columna de la SELECT)
```

NUMCLIE	NOMBCLIE	RECITOTA/12
A001	PEREZ	83.333
A002	SANCHEZ	125
A003	GARCIA	166.666

D.4 MANTENIMIENTO DE DATOS EN LAS TABLAS

D.4.1 Inserción de una fila

Presenta dos formas básicas:

D.4.1.1 Primera posibilidad

```
INSERT INTO nombre-tabla
      (nombres columnas)
VALUES (lista-valores)
```

Ejemplo: Insertar cliente PEREZ número B005 con póliza colectiva número 2003.

```
INSERT INTO POLIZA
      (NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
       RECITOTA, TIPOPOLI)
VALUES (2003,'B005','PEREZ',' ',' ','Colectiva)
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual
2002	B004	MARTINEZ	010192	010192	2500	Colectiva
2003	B005	PEREZ				Colectiva

D.4.1.2 Segunda posibilidad

```
INSERT INTO nombre-tabla
VALUES      (lista-valores)
```

Ignorando las columnas se asume que en VALUES, van en el mismo orden que se definieron.

```
INSERT INTO POLIZA
VALUES (2003,'B005','PEREZ',' ',' ','Colectiva)
```

Con la Primera opción tenemos:

- Independencia de Datos; si creamos una nueva columna entre dos anteriores, el INSERT no dará error.
- Mejor Documentación.

D.4.2 Inserción de múltiples filas

```
INSERT INTO POLIZA
(NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
RECITOTA, TIPOPOLI)
SELECT NUMPOLI, NUMCLIE, NOMBCLIE, FECHALTA, FECHBAJA,
RECITOTA, TIPOPOLI
FROM POLIZA2
```

D.4.3 Actualización

Se realiza a través de la sentencia UPDATE. Su sintaxis es:

```
UPDATE    tabla
SET       nuevos valores de los campos
WHERE     condiciones de actualización
```

Si no se codifica la cláusula WHERE, se actualizan todas las filas.

Ejemplo: Actualizar Recibo Total, multiplicando por 10 los clientes de tipo A*.

```
UPDATE    POLIZA
SET       RECITOTA = RECITOTA * 10
WHERE     NUMCLIE LIKE 'A%'
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	10000	Individual
1002	A002	SANCHEZ	010191	010192	15000	Individual
2001	A003	GARCIA	010191	010192	20000	Individual
2002	B004	MARTINEZ	010192	010192	2500	Colectiva

D.4.4 Borrado

Se realiza a través de la sentencia DELETE. Su sintaxis es:

```
DELETE
FROM    tabla
WHERE    condición
```

o bien

```
DELETE
FROM      tabla
```

Ejemplo:

```
DELETE      *
FROM        POLIZA
WHERE       NUMCLIE LIKE 'B%'
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
1001	A001	PEREZ	010191	010192	1000	Individual
1002	A002	SANCHEZ	010191	010192	1500	Individual
2001	A003	GARCIA	010191	010192	2000	Individual

Ejemplo:

```
DELETE      *
FROM        POLIZA
```

NUMPOLI	NUMCLIE	NOMBCLIE	FECHALTA	FECHBAJA	RECITOTA	TIPOPOLI
---------	---------	----------	----------	----------	----------	----------

APÉNDICE D. INTRODUCCIÓN A SQL	1
D.1 INTRODUCCIÓN. CONCEPTOS PREVIOS	1
D.1.1 Base de Datos Relacional	1
D.1.2 Tabla Relacional	1
D.1.3 El Lenguaje.....	1
D.1.3.1 Selección.....	2
D.1.3.2 Proyección	2
D.1.3.3 Join	3
D.2 SQL. CONCEPTOS BÁSICOS	4
D.2.1 Select Básica.....	4
D.2.2 Concepto de Join.....	5
D.2.3 Actualizar Datos (UPDATE).....	6
D.2.4 Insertar Datos (INSERT).....	6
D.3 SQL AVANZADO.....	7
D.3.1 CRITERIOS DE BUSQUEDA (ORDER BY)	7
D.3.2 OPERADORES	8
D.3.3 EXPRESIONES.....	8
D.3.4 IN y BETWEEN	9
D.3.5 NOT	10
D.3.6 LIKE	10
D.3.7 AND y OR.....	11
D.3.8 JOIN.....	12
D.3.9 DISTINCT.....	12
D.4 MANTENIMIENTO DE DATOS EN LAS TABLAS.....	14
D.4.1 Inserción de una fila.....	14
D.4.1.1 Primera posibilidad.....	14
D.4.1.2 Segunda posibilidad	14
D.4.2 Inserción de múltiples filas.....	15
D.4.3 Actualización	15
D.4.4 Borrado	15