TEMA 6

CREACION, ACTUALIZACION Y CONSULTAS DE BASES DE DATOS UTILIZANDO SQL

108

Introducción

- **SQL** significa **STRUCTURED QUERY LANGUAGE** y es un lenguaje estructurado para realizar consultas.
- SQL es utilizado en muchos **DBMS** para dar mantenimiento y hacer consultas a la base de datos.
- SQL tiene dos componentes principales:
  a) **DDL**: Data Definition Languaje.
  b) **DML**: Data manipulation Languaje.

109

DDL

- Nos permite definir la estructura de una base de datos, definiendo tablas, índices y vistas.
- Algunos comandos soportados por el DDL son los siguientes:

110

Creación de tablas

a) Utilizar el siguiente comando.

```
CREATE TABLE    <nombre_tabla>
( <definicion_columna> [ ,<definicon_columna> ...]
        [ ,< definicion_llave_primaria>]
        [ ,< definicion_atributo_extranjero>
            [ ,< definicion_atributo_extranjero> ... ]
        ]
);
```

111

Creación de tablas

Donde:

<definicion_columna> :: =
nombre_columna tipo [**NOT NULL**]

<defincion_llave_primaria> : : =
PRIMARY KEY (<nombres_columnas>)

<definicion_atributo_extranjero>:: =
FOREIGN KEY (<nombre_columna>)
REFERENCES <nombre_tabla>

112

Ejemplos de creación de tablas

- **Crear la tabla S**

```
CREATE   TABLE   S
   ( S#   CHAR  ( 5)  NOT  NULL,
     SNombre    CHAR   (20)  NOT  NULL ,
     Status INTEGER ,
     Ciudad CHAR  (15),
     PRIMARY   KEY   (S#)
    ) ;
```

113

Ejemplos de creación de tablas

- **Crear la tabla SP**

```
CREATE  TABLE  SP
   ( S#   CHAR  (5)  NOT  NULL,
     P#   CHAR (5)   NOT  NULL,
     CANT  INTEGER,
     PRIMARY  KEY  (S# ,P#),
     FOREIGN  KEY  (S#)  REFERENCES  S,
     FOREIGN  KEY  (P#)  REFERENCES  P
    );
```

114

Otros comandos de tablas

- ***Alterar una tabla*:**
  Utilizaremos el siguiente comando :
  ALTER TABLE <nombre_tabla>
  ADD <nombre_columna> <tipo>;

  Ejemplo:
  ALTER TABLE S **ADD** Teléfono CHAR (15);

- ***Borrando una tabla:***
  Utilizaremos el comando:
  DROP TABLE <nombre_tabla>

115

Comandos para índices

- ***Creación de índices.***

Por medio del comando:

CREATE [UNIQUE] INDEX <nombre_indice>
ON <nombre_tabla> (<nombre_columna> **[A/D]**
[,<nombre_columna> **[A/D]** ..]
);

Ejemplos:

CREATE UNIQUE INDEX Sntx ON S (S#)
CREATE UNIQUE INDEX SPntx ON SP (S#, P#)

116

Comandos para índices

- ***Para borrar un índice*.**

Utilizar la sig. sintaxis:

DROP INDEX <nombre_indice> on <table>

117

DML

- Nos permite realizar **consultas y dar mantenimiento** a la base de datos (altas, bajas y cambios). En realidad es en el DML donde SQL tiene su poderío.

118

- Para los ejemplos que veremos, utilizaremos la siguiente base de datos:

S

S#	NOMBRE	STATUS	CIUDAD
S1	SALAZAR	20	LONDRES
S2	JARAMILLO	10	PARIS
S3	BERNAL	30	PARIS
S4	CAICEDO	20	LONDRES
S5	ALDAMA	30	ATENAS

P

P#	NOMBRE	COLOR	PESO	CIUDAD
P1	TUERCA	ROJO	12	LONDRES
P2	PERNO	VERDE	17	PARIS
P3	TORNI.	AZUL	17	ROMA
P4	TORNI.	ROJO	14	LONDRES
P5	LEVA	AZUL	12	PARIS
P6	RUEDA	ROJO	19	LONDRES

SP

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

119

Operaciones de consulta con el estatuto SELECT.

- La operación básica de consulta en SQL, es el operador SELECT, el cual tiene la siguiente estructura básica.

SELECT <atributo(s)>
FROM <tabla(s)>
[**WHERE** <condición>]

120

Ejemplos de estatuto SELECT

1) Obtenga todos los datos de todos los proveedores.

```
SELECT S#, NOMBRE, CIUDAD,COLOR
FROM S
```

ó

```
SELECT *
FROM S
```

* Sustituye todos los atributos de la tabla S.

121

Ejemplos de estatuto SELECT

2) Obtenga los números de parte de todas las partes suministradas.

```
SELECT DISTINCT P#
FROM SP
```

La cláusula DISTINCT elimina registros duplicados

122

Ejemplos de estatuto SELECT

3) Obtenga números de proveedores con status mayor que 20 y que vivan en PARIS.

```
SELECT S#
FROM S
WHERE (CIUDAD = 'PARIS ') AND
      (STATUS >20 )
```

123

Ejemplos de estatuto SELECT

4) Obtenga número de proveedor y status de proveedores que vivan en PARIS y en orden descendente por STATUS.

```
SELECT S#, STATUS
FROM S
WHERE (CIUDAD = ' PARIS ' )
ORDER BY STATUS DESC
```

La cláusula ORDER BY ordena el resultado

124

Ejemplos de estatuto SELECT

5) Para cada parte suministrada obtenga el número de la parte y los NOMBRES de todas las ciudades que suministran dicha parte.

```
SELECT DISTINCT P# , CIUDAD
FROM SP , S
WHERE ( S.S# = SP.S# )
```

Se establece un producto cartesiano estableciendo a las tablas separadas por coma

125

Ejemplos de estatuto SELECT

6) Obtenga todas las parejas de números de proveedor tal que los dos proveedores estén localizados en la misma ciudad.

| Para renombrar una tabla ponemos el nuevo nombre seguido del nombre original en la parte del FROM

126

Ejemplos de estatuto SELECT

7) Obtenga NOMBRES de proveedor tal que suministren la parte 'p2'.

Una solución es:

```
SELECT    DISTINCT NOMBRE
FROM      SP, S
WHERE     (SP.P# = 'P2')  AND
          (S.S# = SP.S#)
```

127

Ejemplos de estatuto SELECT

- Otra posible solución es utilizando el operador **ANY**, que funciona de la sig. manera:

El operador:

f <Operador-relacional> **ANY** (SELECT...)

Es verdadero si y solo si el valor de f cumple el <operador relacional> (>,<,<>,>=,<=,=) en al menos un valor de los regresados por el SELECT.

128

Ejemplos de estatuto SELECT

El ejercicio 7 utilizando el operador ANY:

```
SELECT  NOMBRE
FROM  S
WHERE   S# = ANY  (SELECT  S#
                   FROM  SP
                   WHERE   P# = ' P2' )
```

129

Ejemplos de estatuto SELECT

8) Obtener números de proveedor, cuyo STATUS sea menor que el valor máximo actual de STATUS.

130

Ejemplos de estatuto SELECT

9) Resolver el mismo problema del ejemplo 7, utilizando el operador IN.

```
SELECT  NOMBRE
FROM    S
WHERE  S#  IN  (SELECT  S#
                FROM  SP
                WHERE   P# ='P2 ')
```

Nota: El operador **= ANY** e **IN** tienen la misma función.

131

Ejemplos de estatuto SELECT

10) Obtenga NOMBRES de proveedor que suministren al menos una parte ROJA.

132

Ejemplos de estatuto SELECT

11) Obtenga NOMBRES de proveedor que suministran la parte p2.

```
SELECT  NOMBRE
FROM  S
WHERE  'P2' IN (SELECT   P#
                FROM     SP
                WHERE  S# = S.S# )
```

Ejemplo de querys anidados correlacionados

133

Ejemplos de estatuto SELECT

12) Obtenga el número de proveedor que suministre al menos una parte suministrada por el proveedor s2.

```
SELECT  DISTINCT  S#
FROM     SP
WHERE   P#  IN  ( SELECT  P#
                  FROM     SP
                  WHERE  ( S# = 'S2'))
```

134

Ejemplos de estatuto SELECT

13) Obténgase números de partes para todas las partes suministradas por más de un proveedor.

135

Ejemplos de estatuto SELECT

14) Obtener NOMBRES de proveedor de aquellos proveedores que no suministren p2. Utilizar el operador ALL.

El operador *** ALL** (donde * es cualquiera de los operadores, =,<>,<,>,>=,<=) se define como sigue la condición :

f *** ALL** (SELECT * ...)

Toma el valor de verdadero si y solo si f, cumple el operador relacional *, en todos los valores regresados por el (SELECT * ...)

136

Ejemplos de estatuto SELECT

```
/* Obtener NOMBRES de proveedor de aquellos
   proveedores que no suministren p2. Utilizar el
   operador ALL. */

SELECT  NOMBRE
FROM  S
WHERE  'P2' <> ALL ( SELECT  P#
                     FROM    SP
                     WHERE    S# = S . S#)
```

137

Ejemplos de estatuto SELECT

15) Obtenga NOMBRES de proveedor que suministren p2 (utilizar el operador EXISTS).

La expresión:

EXISTS (SELECT * ...)

Toma el valor de verdadero si y solo si el resultado de evaluar (SELECT ...) es no vacío, es decir, si SELECT regresa una tabla con al menos un registro.

138

Ejemplos de estatuto SELECT

/* Obtenga NOMBRES de proveedor que suministren p2 (utilizar el operador EXISTS). */

```
SELECT  NOMBRE
FROM  S
WHERE EXISTS (SELECT  *
                    FROM  SP
                    WHERE   S# = S.S#   AND  P# = ' P2')
```

139

Ejemplos de estatuto SELECT

16) Obtenga los NOMBRES de proveedores que suministren todas las partes.

```
SELECT  NOMBRE
FROM  S
WHERE  NOT EXISTS
            (SELECT   *
             FROM   P
             WHERE  NOT  EXISTS
                       (SELECT  *
                        FROM   SP
                        WHERE   S# = S . S#  AND
                                   P# = P .P#))
```

140

Ejemplos de estatuto SELECT

17) Obtenga números de proveedor que suministren al menos todas las partes suministradas por el proveedor s2.

141

Funciones de agregación en SQL

- **a) COUNT:** Cuenta el numero de registros de una tabla.
- **b) SUM:** Suma los valores de una columna.
- **c) AVG:** Calcula el promedio de los valores de una columna.
- **d) MAX:** obtiene el valor mas grande de una columna.
- **e) MIN:** obtiene el valor mas pequeño de una columna.

142

Funciones de agregación en SQL

18) Obtenga el numero total de proveedores.

```
SELECT COUNT(*)
FROM s
```

19) Obtenga el numero total de proveedores que actualmente suministran partes.

```
SELECT COUNT (DISTINCT s#)
FROM sp
```

143

Funciones de agregación en SQL

20) Obtenga la cantidad total de la partes 'P2' suministrada.

```
SELECT SUM(Cantidad)
FROM sp
WHERE p# = 'P2'
```

21) Obtenga números de provedor con status menor que el máximo actual.

```
SELECT s#
FROM s
WHERE status < (SELECT MAX(status)
                FROM s)
```

144

Funciones de agregación en SQL

22) Obtener la cantidad total suministrada por cada una de las partes.

```
SELECT  p#, SUM (cant)
FROM SP
GROUP BY p#
```

145

Funciones de agregación en SQL

- En este query se esta haciendo uso de la función **GROUP BY**. Esta opción permite a SQL agrupar la información respecto a los campos indicados (en el ejemplo respecto a p# perteneciente a SP).
- Cuando se tiene un agrupamiento, los campos seleccionados en el SELECT, se aplican a nivel de grupo, no a nivel de registro. Para el ejemplo por cada grupo se desplegara el numero de p#, y la suma de sus cantidades (el SUM funciona ahora por cada grupo).

146

Funciones de agregación en SQL

23) Obtener los números de partes que sean sumistradas por mas de un proveedor.

```
SELECT p#
FROM SP
GROUP BY p#
HAVING COUNT(*) > 1
```

La cláusula HAVING nos permite seleccionar grupos que cumplan la condición establecida en el HAVING (para nuestro ejemplo grupos con mas de una embarque). Conclusión: WHERE es a registros, como HAVING es a grupos.

147

Operaciones de actualización en SQL.

- El DML de SQL incluye tres operaciones de actualización a una tabla:
  - **INSERT** (insertar)
  - **DELETE** (borrar).
  - **UPDATE** (modificación)

148

Operaciones de actualización en SQL.

24) Inserte la parte 'P7' (de nombre 'MARTILLO', color = 'GRIS', peso = 2, ciudad = 'ATENAS') a la tabla p

```
INSERT INTO p
    VALUES ('P7', 'MARTILLO','GRIS', 2,'ATENAS')
```

25) Borre al proveedor 'S1'.

```
DELETE FROM s
WHERE s# = 'S1'
```

149

Operaciones de actualización en SQL.

26) Borre todos los embarques de los proveedores en Londres.

```
DELETE sp
WHERE 'LONDRES' = (SELECT ciudad
                   FROM s
                   WHERE s# = sp.s#)
```

150

Operaciones de actualización en SQL.

27) Cambie el color de la partes 'p2' a amarillo, aumente se precio en 5 y ponga en NULL su ciudad.

```
UPDATE p
SET color = 'AMARILLO', peso = peso + 5,
    ciudad = NULL
WHERE p# = 'p2'
```

151

Operaciones de actualización en SQL.

28) Ponga la cantidad en cero para todos los proveedores en Londres.

```
UPDATE sp
SET Cantidad = 0
WHERE 'LONDRES' = (SELECT ciudad
                   FROM s
                   WHERE s# = sp.s#)
```

152