

Bases de Datos Unidad 2 Modelo Relacional

Instructor: M.C. Luis Basto Díaz
Email: luisbasto@gmail.com

Modelo Relacional

- Álgebra Relacional
- Lenguaje SQL
- Integridad Referencial
- Normalización
 - Primera Forma Normal
 - Segunda Forma Normal
 - Tercera Forma Normal
 - Forma Normal de Boyce - Codd
 - Criterios de Normalización

Creación de tablas

- Para crear tablas se deberá proveer el nombre de la tabla y los tipos de datos de las columnas:

```
CREATE TABLE                               USE Northwind
Table_name (                               CREATE TABLE Drivers (
column_1 data_type,                       license VARCHAR(15),
column_2 data_type,                       firstname VARCHAR(30),
.                                         lastname VARCHAR(30)
.                                         )
column_n data_type
)
```

M.C. Luis R. Basto Díaz, CEL

3

Creación de tablas

- Cuando una tabla es creada automáticamente adiciona una nueva fila a la tabla *sysobjects* y una nueva fila es adiciona a *syscolumns* por cada columna en la tabla.
- Para permitir/no permitir valores nulos en los campos, SQL provee de la sig. Sintaxis:

```
USE Northwind

CREATE TABLE Cars (
    serial VARCHAR(200) NOT NULL,
    make VARCHAR(100) NOT NULL,
    model VARCHAR(100) NOT NULL,
    color VARCHAR(50) NULL
)
```

M.C. Luis R. Basto Díaz, CEL

4

Creación de tablas

- Las tablas pueden contener claves primarias, las cuales pueden ser una columna o un conjunto de columnas que identifican de manera única una fila de la tabla:

```
CREATE TABLE Agente(
  Agenteid INT IDENTITY(10,1) NOT
  NULL,
  Agentename VARCHAR(100) NOT
  NULL,
  Direccion VARCHAR(200) NULL,
  Ciudad VARCHAR(100) NULL,
  Estado CHAR(2) NULL,
  CP VARCHAR(10) NULL
)
GO
```

Se crea una clave primaria (Agenteid), con una semilla de 10 y con incrementos de 1 en 1.

M.C. Luis R. Basto Díaz, CEL

5

Modificando tablas

- Para modificar (eliminar/adicionar) una tabla previamente creada, se tiene los siguientes comandos:

```
USE Northwind
ALTER TABLE Cars ADD mileage INT
NULL
GO
```

```
USE Northwind
ALTER TABLE Cars DROP COLUMN
mileage,color
GO
```

M.C. Luis R. Basto Díaz, CEL

6

Actualizaciones de la BD

- Para modificar los contenidos de la BD, SQL maneja tres instrucciones:
 - INSERT
 - UPDATE
 - DELETE

M.C. Luis R. Basto Díaz, CEL

7

Adición de datos a la BD

- INSERT tiene dos formas, una donde puede insertarse una sola fila en la BD.
 - `INSERT INTO tabla [(ListaColumnas)] VALUES (ListaValores).`
 - ListaColumnas, es la lista de una o más columnas separadas por comas.
 - ListaColumnas es opcional, si se omite, SQL tomará toda la lista de columnas existentes en la tabla.
 - ListaValores, es la lista de los valores que tendrán las columnas.

M.C. Luis R. Basto Díaz, CEL

8

- Algunas consideraciones para la instrucción anterior son:
 - El número de elementos de cada lista debe ser el mismo.
 - Debe haber una correspondencia entre los elementos en ambas listas, el primer elemento de la lista se aplica al primer elemento de los valores y así sucesivamente.
- El tipo de dato de cada columna debe ser compatible con el tipo de dato de la correspondiente columna.

M.C. Luis R. Basto Díaz, CEL

9

- **INSERT INTO tabla VALUES ('valor1', 'valor2', 'valor3', 'valor4', 'valor5', 'valor6').**
- **INSERT INTO tabla (col1,col2,col3,col4) VALUES ('valor1', 'valor2', 'valor3', 'valor4').**
- **INSERT INTO tabla VALUES ('valor1', 'valor2', NULL, NULL).**

M.C. Luis R. Basto Díaz, CEL

10

Modificación de datos en la BD (Update)

- La instrucción UPDATE permite modificar el contenido de filas ya existentes en una tabla especificada.
- UPDATE NombreTabla SET col1 = 'valor1', col2 = 'valor2'... [WHERE condBúsqueda]
- UPDATE tabla SET col1 = col1*1.2;
- UPDATE tabla SET col1 = col1*1.2 WHERE col2 = 'valor1';
- UPDATE tabla SET col1 = col1*1.2, col2 = '1500' WHERE col3 = 'valor1';

M.C. Luis R. Basto Díaz, CEL

11

Borrado de datos (DELETE)

- La instrucción DELETE permite borrar filas de una tabla especificada.
- DELETE FROM tabla [WHERE condBúsqueda];
- DELETE FROM tabla WHERE col1='valor1';
- DELETE FROM tabla WHERE col1 > '5000';
- DELETE FROM tabla;

M.C. Luis R. Basto Díaz, CEL

12

Operaciones con SQL

- Transacciones
 - Commit, Rollback
- Procedimientos almacenados
- Triggers
- Manejo de Vistas

M.C. Luis R. Basto Díaz, CEL

13

Manipulación de datos

- DDL: Lenguaje de definición de datos.
 - Define la estructura de la base de datos y controlar el acceso a los datos.
- DML: Lenguaje de manipulación de datos
 - Extrae y actualiza los datos.
- DCL: Lenguaje de Control de Datos
 - Otorga y cancela privilegios a la BD

M.C. Luis R. Basto Díaz, CEL

14

Manipulación de datos

- Se examina las instrucciones DML de SQL:
 - SELECT
 - INSERT
 - UPDATE
 - DELETE

M.C. Luis R. Basto Díaz, CEL

15

Consultas simples

- SELECT
 - El propósito es extraer y visualizar datos de una o más tablas de la base de datos.
 - Es un comando capaz de realizar el equivalente de las operaciones de selección, proyección y combinación
- SELECT [DISTINCT | ALL]{* | [expresiónColumna[AS nuevoNombre]][,..]}
FROM NombreTabla [alias] [,...]
[WHERE condición]
[GROUP BY listacolumnas][HAVING condición]
[ORDER BY listacolumnas]

M.C. Luis R. Basto Díaz, CEL

16

Consultas simples

- `Expresioncolumna` representa un nombre de columna o una expresión.
- `NombreTabla` es el nombre de una tabla o vista de la BD.
- `Alias` es una abreviatura opcional para `NombreTabla`.

M.C. Luis R. Basto Díaz, CEL

17

Consultas simples

- `FROM`: Especifica la tabla o tablas que hay que usar.
- `WHERE`: Filtra las filas de acuerdo con alguna condición.
- `GROUP BY`: Forma grupos de filas que tengan el mismo valor de columna.
- `HAVING`: filtra los grupos de acuerdo con alguna condición.
- `SELECT`: Especifica qué columnas deben aparecer en la salida
- `ORDER BY`: Especifica el orden de la salida.

M.C. Luis R. Basto Díaz, CEL

18

Extracción de filas

- Extraer todas las columnas de todas las filas
 - `SELECT col1,col2,...,coln FROM tabla;`
 - `SELECT * FROM tabla;`
- Extraer una serie de columnas específicas de todas las filas.
 - `SELECT col1,col5,col7 FROM tabla;`

M.C. Luis R. Basto Díaz, CEL

19

Uso de Distinct

- Generar un listado con filas no duplicadas
 - `SELECT DISTINCT col1,col2,...,coln FROM tabla;`
- Se obtiene la tabla de resultados en la que los elementos (filas) duplicados se eliminan.

M.C. Luis R. Basto Díaz, CEL

20

Campos calculados

- Campo calculado o derivado:
 - `SELECT col1,col2,col3, col_res_op_aritmética FROM tabla;`
 - `SELECT col1,col2,col3, col_res_op_aritmética AS nombreColumna FROM tabla;`
- Operaciones aritméticas: +, -, *, /.
- Los campos calculados permiten obtener un campo más mediante una operación aritmética. El resultado es un campo con un nombre genérico.
- Si se le agrega la sentencia `AS` con un nombre específico, la columna tendrá ese nombre proporcionado.

M.C. Luis R. Basto Díaz, CEL

21

Selección de filas: Cláusula WHERE

- Las cinco condiciones básicas de búsqueda son las siguientes:
 - Comparación: Compara el valor de una expresión con el valor de otra.
 - Rango: Comprueba si el valor de una expresión cae dentro de un rango especificado de valores.
 - Pertenencia a conjunto: Comprueba si el valor de una expresión coincide con uno de los valores de un cierto conjunto.

M.C. Luis R. Basto Díaz, CEL

22

- Condición de búsqueda basada en comparación:
 - `SELECT col1,col2,col3, FROM tabla WHERE col3>dato;`
- Operadores de comparación disponibles:

=	Igual
<>	Distinto
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
!=	Distinto en algunos dialectos

M.C. Luis R. Basto Díaz, CEL

23

Comparación compuesta

- Pueden utilizarse operadores lógicos AND, OR y NOT.
 - `SELECT * FROM tabla WHERE col1 = dato1 OR col2=dato2;`
 - `SELECT * FROM tabla WHERE col1 = dato1 AND col2=dato2;`

M.C. Luis R. Basto Díaz, CEL

24

Búsqueda basada en rango

- Se utiliza para un rango de valores:
 - `SELECT * FROM tabla WHERE col1 BETWEEN dato1 AND dato2;`
 - `SELECT * FROM tabla WHERE col1 NOT BETWEEN dato1 AND dato2;`

M.C. Luis R. Basto Díaz, CEL

25

Búsqueda basada en pertenencia de conjuntos (IN/NOT IN)

- `SELECT * FROM tabla WHERE col1 IN ('dato1','dato2','dato3',...);`
- Comprueba si el valor de los datos se corresponde con uno de los valores especificados en una determinada lista.
- La condición IN se puede escribir de la siguiente manera:
 - `SELECT * FROM tabla WHERE col1 = 'dato1' OR col2 = 'dato2', OR 'dato3' ...;`

M.C. Luis R. Basto Díaz, CEL

26

Búsqueda basada en correspondencia de patrones

- LIKE/NOT LIKE
- El carácter de porcentaje (%) representa cualquier secuencia de cero o más caracteres (comodín).
- El carácter guión bajo (_) representa cualquier carácter individual.
- Ejemplos:
 - Dirección LIKE 'M%'
 - Dirección LIKE '%a'
 - Dirección LIKE 'H_%'
 - Dirección NOT LIKE 'M%'
- `SELECT * FROM tabla WHERE dirección LIKE 'M%';`

M.C. Luis R. Basto Díaz, CEL

27

Ordenación de resultados

- La ordenación se puede realizar de forma ascendente y descendente.
 - `SELECT * FROM tabla ORDER BY col DESC;`
 - `SELECT * FROM tabla ORDER BY col ASC;`
- Se puede ordenar un resultado mediante varias columnas.
 - `SELECT * FROM tabla ORDER BY col1,col2,col3... DESC;`
 - `SELECT * FROM tabla ORDER BY col1,col2,col3... ASC;`

M.C. Luis R. Basto Díaz, CEL

28

Utilización de funciones de agregación

- **COUNT**: devuelve el número de valores en una columna especificada.
- **SUM**: devuelve la suma de los valores contenidos en una columna especificada.
- **AVG**: devuelve la media de los valores contenidos en una columna especificada
- **MAX**: devuelve el valor máximo contenido en una columna especificada.
- **MIN**: devuelve el valor mínimo contenido en una columna especificada.

M.C. Luis R. Basto Díaz, CEL

29

- **SELECT col1,col2 COUNT(col3) FROM tabla;**
- **SELECT COUNT(*) AS miCount FROM tabla WHERE col1 > 1000;**
- **SELECT COUNT(DISTINCT col1) AS miCount FROM tabla WHERE col1 BETWEEN 1 AND 1000;**
- **SELECT COUNT(DISTINCT col1) AS miCount FROM tabla WHERE col1 BETWEEN 1 AND 1000;**
- **SELECT COUNT(col1) AS miCount, SUM(col2) AS miSum FROM tabla WHERE col1 = 'valor';**

M.C. Luis R. Basto Díaz, CEL

30

Agrupación de resultados

- A menudo resulta útil disponer de subtotales en los informes, en SQL esto se realiza mediante la cláusula GROUP BY.
- `SELECT col1, COUNT(col2) AS miCount, SUM(col3) AS miSum FROM tabla GROUP BY col1 ORDER BY col1;`

M.C. Luis R. Basto Díaz, CEL

31

Subconsultas

- Los resultados de una instrucción SELECT pueden utilizarse dentro de otra instrucción SELECT para determinar el contenido de la tabla final.
- `SELECT col1,col2,col3 FROM tabla1 WHERE col4 = (SELECT col4 FROM tabla2 WHERE col1 = 'Valor');`

M.C. Luis R. Basto Díaz, CEL

32

Reglas a las subconsultas

- No puede utilizarse la cláusula ORDER BY dentro de una subconsulta.
- La lista SELECT de la subconsulta debe estar compuesta por un único nombre de columna o expresión.

M.C. Luis R. Basto Díaz, CEL

33

Tipos de integridad en los datos

- Las BD son útiles por la calidad de los datos que ellos contienen.
- La calidad de los datos está determinado por una serie de factores.
- El diseño lógico de la BD, la implementación física, las aplicaciones de cliente, y el usuario final tienen un papel importante en la calidad de los datos.

M.C. Luis R. Basto Díaz, CEL

34

Tipos de integridad en los datos

- La integridad de los datos es un factor importante que contribuye a la calidad completa de los datos.
- SQL Server provee diferentes maneras de implementar la integridad de los datos.

M.C. Luis R. Basto Díaz, CEL

35

Tipos de integridad en los datos

- Consideremos una BD en la cual se almacena información sobre productos, clientes, ventas, etc.
 - ¿Está la información relacionada a un producto específico de una manera consistente?
 - ¿Se tiene diferentes productos con el mismo nombre o código?
 - ¿Puedes identificar a los clientes de una manera única, aún si ellos tienen el mismo nombre?
 - ¿Hay algunas ventas que no están relacionadas con algunos clientes?
 - ¿Se tienen algunas ventas de algunos productos no existentes?

M.C. Luis R. Basto Díaz, CEL

36

Tipos de integridad en los datos

- Para asegurar la integridad de los datos contenidos en una BD se debe asegurar:
 - Todo valor individual se adapte a las reglas de negocio específicas (integridad de dominio).
 - Todo objeto puede ser identificado de manera única y de forma inequívoca (integridad de entidad).
 - Los datos relacionados deben estar correctamente conectados (integridad relacional)

M.C. Luis R. Basto Díaz, CEL

37

Integridad de dominio

- Al aplicar reglas de negocio para validar los datos almacenados en la BD impone la integridad de dominio.
 - El tipo de dato de la columna restringe los valores que se pueden introducir en la columna
 - La longitud de la columna impone la longitud de los datos que introduces en una columna específica.
 - Se puede forzar a la longitud mínima o máxima de un valor dado.
 - Se puede restringir los datos a un formato adecuado.
 - Puede ser útil restringir el rango de valores validos

M.C. Luis R. Basto Díaz, CEL

38

Integridad de entidad

- Cada objeto real debería ser fácilmente identificado en una BD.
 - Debería ser fácil identificar a un cliente por un único valor simple, tal como: 1234, almacenado en una columna de identificación Clienteld.

M.C. Luis R. Basto Díaz, CEL

39

Integridad referencial:

- Esta regla se aplica a las claves externas.
- Si hay una clave externa en una relación, el valor de la clave externa debe corresponderse con el valor de una clave candidata de alguna tupla en su relación de origen o el valor de la clave externa debe ser completamente nulo.

M.C. Luis R. Basto Díaz, CEL

40

NoSucursal	Calle	Ciudad	CodPostal
B005	22 México Norte	Monterrey	97200
B007	32 Miraflores	Campeche	95200
B003	17 Privada del norte	Mérida	93250
B004	56 Calle azul	Saltillo	97249
B002	67 Doctores	Monterrey	98600

NoEmp	NomEmp	ApEmp	Posición	Sexo	FechaNac	Salario	NoSucursal
SL21	Juan	Blanco	Admin.	M	1-oct-76	15000	B005
SG37	Ana	Castillo	Asistente	F	2-ene-70	6500	B003
SG14	David	Díaz	Supervisor	M	8-dic-77	8900	B003
SA9	Maria	Canto	Asistente	F	7-may-72	6500	B007
SG5	Susana	Solís	Admin.	F	5-abr-79	15000	B003
SL41	Julia	Soto	Asistente	F	2-mar-74	6500	B005

M.C. Luis R. Basto Díaz, CEL

41