

IN3501 – Tecnologías de Información y Comunicaciones para la Gestión
Primavera 2009

SQL: Structured Query Language

Universidad de Chile
Departamento de Ingeniería Industrial

Auxiliares:

Iván Videla (ividela@dcc.uchile.cl)

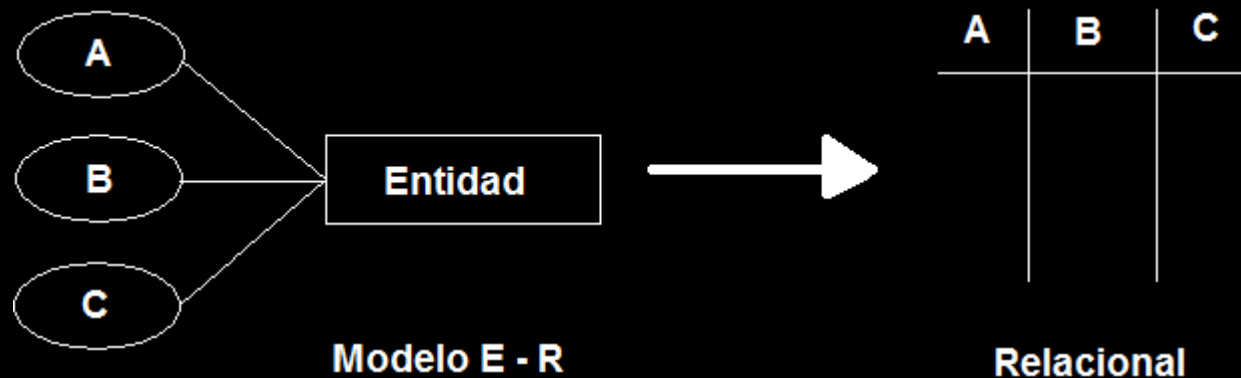
Claudio Millán (cmillan@dcc.uchile.cl)

Modelo Relacional

- ❑ El modelo relacional es una forma de ver los datos es decir, es una receta para representar los datos, mediante tablas, y la receta para manipular esa representación mediante operadores .
- ❑ El modelo relacional se preocupa de tres aspectos de los datos : su estructura, su integridad y su manipulación.
- ❑ Con la introducción del modelo relacional se desarrolló una teoría orientada a las bases de datos relacionales. Esta teoría ayuda al diseño de las bases de datos y al proceso de consultas del usuario

Modelo Relacional

- ❑ En este modelo la base de datos es vista por el usuario como una relación de tablas. Cada fila de una tabla es un registro o tupla y los atributos son columnas o campos.



Llave Primaria

- ❑ Aseguran la NO existencia de filas duplicadas.
- ❑ Se trata de un atributo o conjunto de ellos, que definen en forma única a una fila.
- ❑ Implícitamente, definen una estructura de datos que permita asegurar la NO repetición de datos.

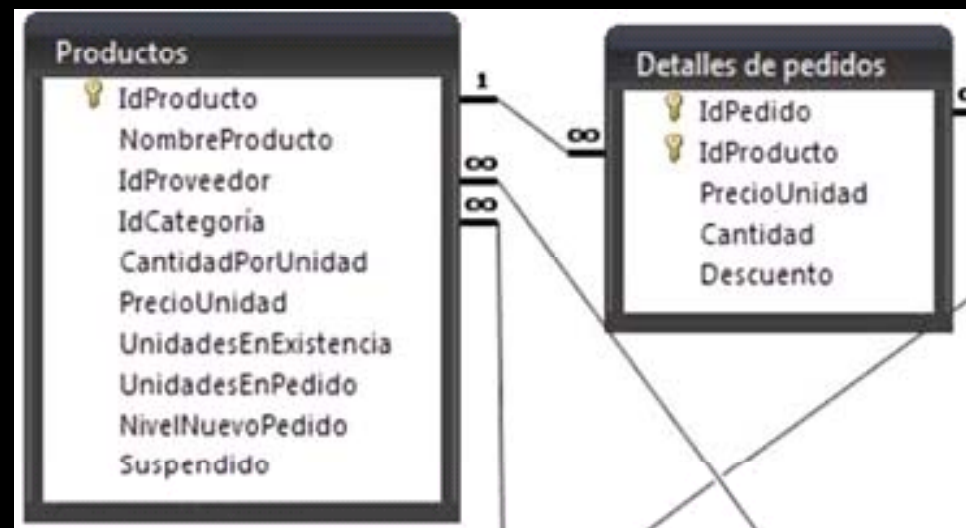
Ejemplo: En la entidad Clientes, el atributo IdCliente identifica en forma única a algún cliente



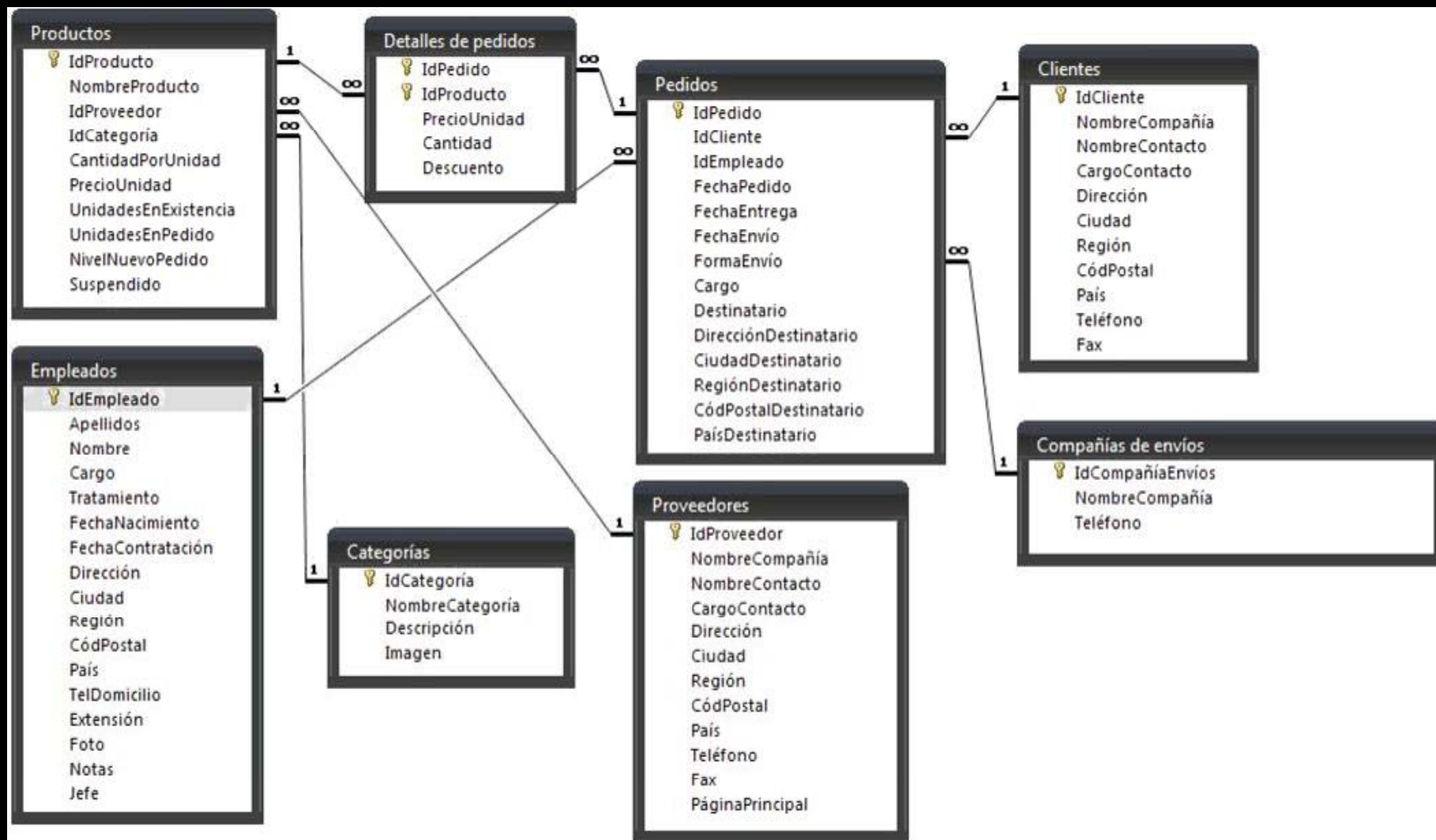
Llave foránea (Foreign Key)

- ❑ Es una columna o conjunto de estas, que referencian a una PK o UK en la misma tabla o en otra.
- ❑ Sirven para definir relaciones tales como las master-detail.

Ejemplo : No se puede borrar un maestro, sin haber borrado primero el detalle, no puedo borrar un producto si es que existe un detalle pedido.



Modelo Relacional



Objetivos de SQL

- ❑ **Un lenguaje de base de datos debe permitir :**
 - Crear las estructuras de la Base de datos y sus relaciones.
 - Realizar inserción, modificación y borrado de datos.
 - Realizar consultas simples y complejas
- ❑ **Estas tareas se deben realizar con un mínimo esfuerzo del usuario.**
- ❑ **La estructura y sintaxis de comandos debe ser fácil de aprender.**
- ❑ **Debe ser portable**

Propósito e Importancia de SQL

❑ Recuperar datos desde una Base de Datos usando **SELECT** :

- Uso de la componente **WHERE** para agregar condiciones a las Consultas.
- Ordenar los resultados de las Consultas usando **ORDER BY**.
- Agrupar datos usando **GROUP BY**
- Poniendo condiciones sobre los grupos de datos usando **HAVING**.
- Uso de Sub-Consultas.

Sentencia SELECT

```
SELECT [DISTINCT | ALL]  
{* | [column_expression [AS new_name]] [,...]}  
FROM table_name [alias] [, ...]  
[WHERE condition]  
[GROUP BY column_list]  
[ HAVING condition]  
[ORDER BY column_list]
```

SELECT Especifica las columnas (atributos) que se proyectarán en la consulta.

FROM Especifica las tablas que se usaran para la consulta.

WHERE Establece los filtros sobre las tuplas seleccionadas.

GROUP BY Permite agrupar los datos. Esta columna debe estar incluida en el SELECT.

HAVING Permite generar filtros sobre los grupos de datos.

ORDER BY Especifica el orden de los datos.

Otros comandos para SELECT

- ❑ **LIKE**: Permite comparar expresiones regulares entre dos variables.
- ❑ **SUM**: Función de agregación que permite calcular la suma de un conjunto de elementos numéricos.
- ❑ **AVG**: Función de agregación que permite calcular la promedio de un conjunto de elementos numéricos.
- ❑ **COUNT**: Función de agregación que permite contar un conjunto de elementos numéricos.
- ❑ **CONCAT(A,B)** : Función que concatena A con B, retornando la variable de tipo “varchar” AB.

Otros comandos para SELECT

- ❑ **IF(*condicion,then,else*)**: Función lógica que permite evaluar una condición lógica, entregando como resultado *then* en caso de ser verdadera, o *else* en caso de ser falsa.
- ❑ **YEAR(date)**: Función que retorna el año asociado a una determinada variable de tipo date.
- ❑ **MONTH(date)**: Función que retorna el mes asociado a una determinada variable de tipo date.
- ❑ **numero 1 <> numero 2**: Operador de desigualdad entre el numero 1 y el numero 2.
- ❑ **DISTINCT** para eliminar los códigos duplicados:

Inserción de datos, INSERT

- ❑ **INSERT INTO** table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
- ❑ Si se omite la lista de columnas SQL asume que son todas las columnas en el orden correspondiente.

Actualización (UPDATE) y borrado de datos (DELETE)

- ❑ **UPDATE** table_name
 Set column_name1 = data_value1
 [, column_name2 = data_value2...]
 [**WHERE** search_condition]
- ❑ **DELETE FROM** table_name
 [**WHERE** search_conditio]

Preguntas

- ❑ Una lista con las empresas, el nombre de contacto, el cargo y el teléfono de aquellos clientes cuyos contactos sean propietarios o gerentes.
- ❑ Una lista con los destinatarios y el monto de sus pedidos para el año 1996.
- ❑ El promedio sobre todos los pedidos de los destinatarios para el año 1996.
- ❑ Que me indique las compañías que tienen la palabra Ana en su nombre como tipo 1.
- ❑ Una lista con la cantidad de pedidos con descuento, agrupados por el semestre de cada año.