

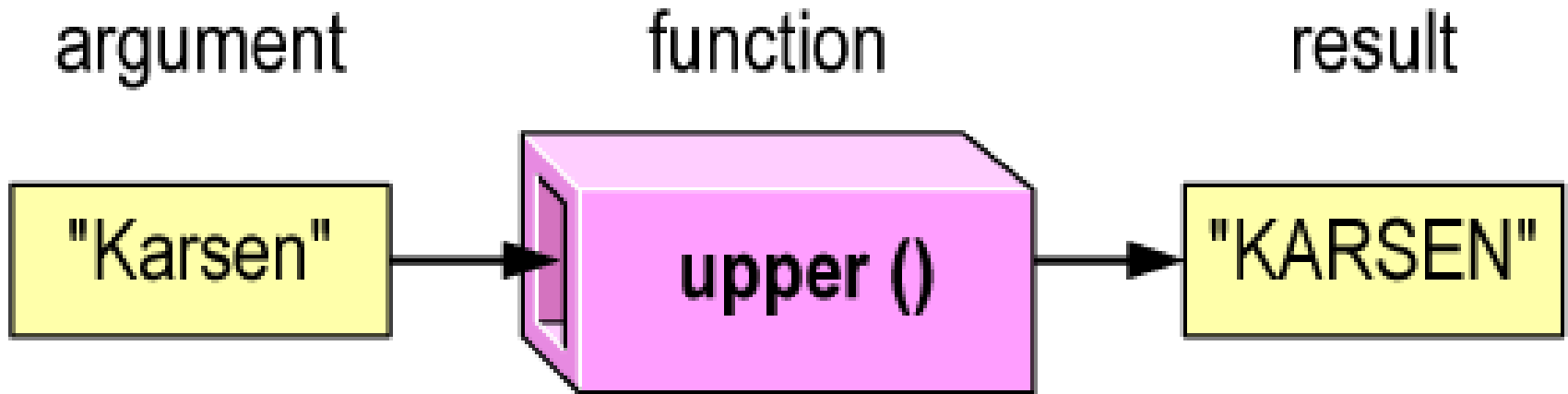
# Bases de Datos Relacionales

## FUNCIONES

Preparó: *Ismael Castañeda Fuentes*  
Fuentes: Manuales Sybase  
Manuales SQL Server  
Manuales Oracle

# Function (funciones)

- Una function es una operación que manipula datos de una determinada manera



- Terminología
  - Argumento – Valor o expresión dada a la function
  - Resultado – Valor o expresión retornado por la function

# Uso de funciones

- Las funciones se pueden usar en:
  - Listas de un **select**
  - Cláusulas **where**
  - En cualquier sitio donde se permita una expresión

# Uso de funciones en listas de un select

- Sintaxis simplificada:

`select function_name (arguments)`

- Ejemplo:

```
select distinct upper(type) from titles
```

```
-----  
BUSINESS  
MOD_COOK  
TRAD_COOK  
UNDECIDED  
PSYCHOLOGY  
POPULAR_COMP
```

# Uso de funciones en una cláusula where

- Sintaxis simplificada:

```
select column_list  
  from table_name  
  where condition_with_one_or_more_functions
```

- Ejemplo:

```
select title from titles where lower(title) like "%the%"
```

```
title
```

```
-----
```

```
The Gourmet Microwave
```

```
The Psychology of Computer Cooking
```

```
(2 rows affected)
```

# Conversión de datatypes

- Bajo muchas circunstancias, los servidores pueden comparar dos valores de diferentes datatypes
- Conversión implícita
  - La que hace automáticamente el servidor
- Conversión explícita
  - Conversión que requiere de la función **convert**
- Conversión no soportada
  - Conversión que no puede realizar el servidor

# Función convert

- La función **convert** cambia valores de un datatype a otro

- Sintaxis simplificada:

```
convert (datatype, expression [, style ] )
```

- Ejemplo\_1 Se convierte *money* a *char(10)*:

```
select price from titles where convert (char(10), price) like "%.99"
```

- Ejemplo\_2

- Verificar la base de datos en que se trabaja:

```
select db_name()
```

- Escribir esta consulta usando "+" (requiere operandos tipo string. La consulta fallará:

```
select "The price of "+ title + " is $" + price  
from pubs2..titles
```

- ¿Por qué falla la consulta?

- Reescribir la sentencia usando **convert**:

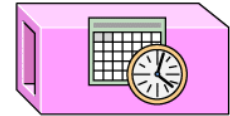
```
select "The price of "+ title + "is $" +  
convert(varchar(10),price)  
from pubs2..titles
```

# Categorías de funciones

- Date
- Mathematical
- String
- Aggregate



# Funciones date



- Las funciones *Date* son extensiones SQL que manipulan datos tipo *datetime*

- Ejemplos

- Función **getdate**

```
select getdate()  
-----  
Feb 4 1999 12:00 AM
```

- Función **datetime**

```
select datetime(mm, pubdate)  
from titles  
where title = "Net Etiquette"  
-----  
July
```

- Función **dateadd**

```
select dateadd(dd, 5, pubdate)  
from titles  
where title = "Net Etiquette"  
-----  
Jul 29 1991 12:00 A
```

# Funciones matemáticas



- Las funciones matemáticas son extensiones SQL que manipulan datos numéricos

- Ejemplos

- Función **round**

```
select round(price, 0)
from titles
where title_id = "PC1035"
-----
23.00
```

- Función **floor**

```
select floor(3.14159)
-----
3
```

- Función **sqrt** (raíz cuadrada)

```
select sqrt(122)
-----
11.045361017187261
```

# Funciones string



- Las funciones de string son extensiones SQL que manipulan datos de caracteres
- Ejemplos

- Función **substring** :

```
select substring("(510) 922-4087",7,8)
-----
922-4087
```

- Función **right** :

```
select right("Mr. Ringer", 6)
-----
Ringer
```

- Función **lower** :

```
select au_lname, au_fname from authors
       where lower(au_lname) like ("de%")
au_lname          au_fname
-----          -----
DeFrance          Michael
del Castillo      Innes
```

# Operador +

- Funcionalmente similar a una función de string
- Concatena dos o más strings de caracteres

- Ejemplo\_1:

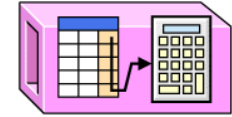
```
select "Mr. " + "Harry Sullivan"
-----
Mr. Harry Sullivan
```

- Ejemplo\_2:

```
select au_id, au_lname + ", " + au_fname as "name" from authors

au_id          name
-----
172-32-1176    White, Johnson
213-46-8915    Green, Marjorie
238-95-7766    Carson, Cheryl
...
```

# Funciones aggregate



- Las funciones **Aggregate** son estándares ANSI que ejecutan operaciones matemáticas con valores de las columnas
- Excepto **count(\*)**, las funciones **aggregate** ignoran los NULLs
- No se pueden usar en una cláusula **where**, excepto si hace parte de un subquery

```
select title_id, price
from titles
where price > (select avg(price) from titles)
```

- función **count(\*)**

```
select count(*) from titles
where type = "popular_comp"
-----
3
```

- función **count(column\_name)**

```
select count(price) from titles
where type = "popular_comp"
-----
2
```

- función **avg**

```
select avg(price) from titles
where type = "popular_comp"
-----
21.48
```

# Función isnull

- La función **isnull** reemplaza valores tipo NULL en un determinado valor no NULL

- Sintaxis:

**isnull** (*column\_which\_may\_have\_NULL\_values, non-NULL value*)

- Ejemplo:

```
select avg(price) from titles
```

```
-----
```

```
14.77
```

```
select avg(isnull (price, $0.00)) from titles
```

```
-----
```

```
13.13
```