



Arquitectura y Administración de bases de datos con SQL 2021

Héctor Manuel Garduño Castañeda

Diciembre, 2021



Contenido

Piso y techo

Aleatoriedad

Semilla de aleatoriedad

Redondeo

Potencias



Las funciones *piso* y *techo* en matemáticas se definen como el mayor entero que es menor que un número, y el menor entero que es mayor a ese número. Se denotan por $\lfloor \cdot \rfloor$ y $\lceil \cdot \rceil$. Por ejemplo, $\lfloor 3.4 \rfloor = 3$ y $\lceil 3.4 \rceil = 4$. En pgadmin son las funciones **FLOOR** y **CEIL** cuya sintaxis es **FLOOR**($\langle\langle \text{número} \rangle\rangle$) y **CEIL**($\langle\langle \text{número} \rangle\rangle$).

```
SELECT FLOOR(3.4);
```

```
SELECT CEIL(3.4);
```

- Versión puntual

```
SELECT order_line, sales, FLOOR(sales), CEIL(sales)
```

```
FROM sales
```

```
WHERE discount > 0;
```

-Versión vectorizada



+ Nunca devuelve al uno

+ Esto nos permite simular cualquier dist de probabilidad

La función **RANDOM** puede ser usada para generar un número aleatorio entre 0, inclusive, y 1, exclusive. Es decir, simula una distribución uniforme en $[0, 1)$. Es importante notar que no lleva argumentos, por lo que su sintaxis es simplemente **RANDOM()**.

Para generar un número aleatorio en el intervalo $[a, b)$ hacemos
 $\text{RANDOM()} * (b - a) + a$

Para generar un entero aleatorio en el intervalo $[a, b]$ hacemos (este si considera al valor b)
 $\text{FLOOR}(\text{RANDOM()} * (b - a + 1)) + a$. En particular,

● $\text{FLOOR}(\text{RANDOM()} * m) + 1$ genera un número aleatorio en el conjunto $\{1, 2, \dots, m\}$.

+ Útil para lanzamiento de un dado

+ Se hace un desplazamiento con +a y luego del (b-a) se toma un porcentaje (ie entre 0 y 1) de la longitud del intervalo



```
SELECT RANDOM(), RANDOM()*(50-10)+10,  
FLOOR(RANDOM()*(50-10+1))+10;
```

Problema. Vas a ir a encuestar 10 casas, de las cuales no conoces el número de habitantes en cada una. Sabes que vas a seleccionar a una persona aleatoriamente en cada casa. ¿Cómo resuelves el problema?



Si queremos fixar el mismo número aleatorio para, por ejemplo, repetir las simulaciones y obtener el mismo resultado, debemos fijar una semilla de aleatoriedad. Esto se hace con la función **SETSEED**, cuya sintaxis es **SETSEED(⟨semilla⟩)**, donde 'semilla' es un número entre -1 y 1, ambos inclusive.

```
SELECT SETSEED(0.5);  
SELECT RANDOM(), RANDOM()*(50-10)+10,  
FLOOR(RANDOM()*(50-10+1))+10;
```

Observación. Es muy importante notar que el setseed únicamente aplicará al primer comando que contenga aleatoriedad.

+ Cuando fijamos una semilla y ejecutamos el código en conjunto (las dos líneas) lanzará el mismo número aleatorio

+ Si solo ejecutamos la segunda línea, generará otro número aleatorio



La función **ROUND** redondea números hasta una cierta cantidad de decimales. Su sintaxis es **ROUND**(⟨⟨número⟩⟩,⟨⟨cantidad de decimales⟩⟩).
Si la cantidad de decimales se omite, hará el redondeo hacia enteros.

```
SELECT order_line, sales, ROUND(sales),ROUND(sales,2)  
FROM sales;
```

+ A partir del 0.5 redondea al próximo entero



La función **POWER** sirve para calcular potencias. Su sintaxis es **POWER**(m, n), que equivale a m^n .

```
SELECT POWER(6,2), POWER(4,0.5);
```

```
SELECT age, POWER(age,2)  
FROM customer ORDER BY age;
```

