

# Ayudantía 2

---

BASES DE DATOS

PROFESOR: SEBASTIAN ALVARADO

AYUDANTE: VICENTE DIAZ

SECCIÓN 1



# Contacto

---

Discord: BarrosLucoPalta

WSP: +569 5749 6014

Steam: Coustillier

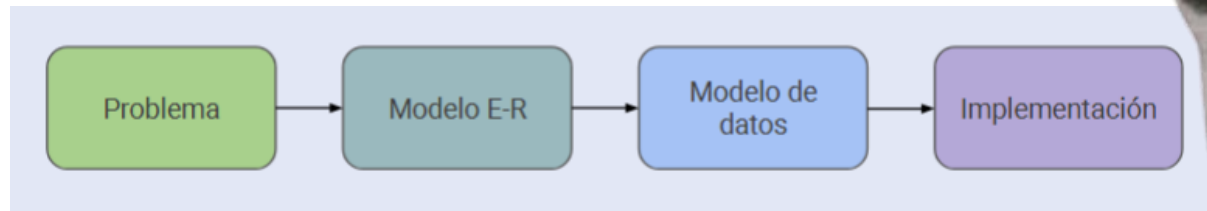


# ¿Donde nos encontramos?

---

Antes de seguir avanzando debemos conocer los tipos de datos que existen:

- Tipos de datos numéricos
- Tipos de datos de texto y caracteres
- Tipos de datos de fecha y hora
- Tipos de datos booleanos
- Tipos de datos de identificadores únicos
- Tipos de datos de colecciones y estructuras
- Tipos de datos binarios



# Tipos de datos numéricos

---

- **INTEGER:** números enteros en el rango de -2147483648 a 2147483647.
- **BIGINT:** números enteros en un rango mucho más amplio, de -9223372036854775808 a 9223372036854775807.
- **NUMERIC/DECIMAL:** números de punto flotante con precisión decimal arbitraria. Puede definirse con una precisión y una escala para controlar el número de dígitos permitidos en total y después del punto decimal.
- **REAL:** Almacena números de punto flotante de precisión simple.
- **DOUBLE PRECISION:** Almacena números de punto flotante de precisión doble.



Tipo de datos	Intervalo	Expresión de intervalo	Storage
bigint	De -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	$-2^{63}$ a $2^{63}-1$	8 bytes
int	De -2.147.483.648 a 2.147.483.647	$-2^{31}$ a $2^{31}-1$	4 bytes
smallint	De -32 768 a 32 767	$-2^{15}$ a $2^{15}-1$	2 bytes
tinyint	De 0 a 255	$2^{0}-1$ a $2^{8}-1$	1 byte

# Tipos de datos de texto y caracteres

---

- CHARACTER/CHAR: Almacena cadenas de caracteres de longitud fija. Debes especificar la longitud máxima.
- VARCHAR: Almacena cadenas de caracteres de longitud variable. Puedes especificar la longitud máxima.
- TEXT: Almacena cadenas de caracteres de longitud variable, similar a VARCHAR, pero sin una longitud máxima definida.



## String

Data Structure

```
string str = "Geeks"
```

index	→	0	1	2	3	4	5
str	→	G	e	e	k	s	\0

# Tipos de datos de fecha y hora

---

- DATE: Almacena solo la fecha (año, mes y día). Ejemplo: 2025-09-07.
- TIME: Almacena solo la hora (horas, minutos y segundos). Ejemplo: 14:35:22.
- DATETIME: Almacena fecha y hora completas. Ejemplo: 2025-09-07 14:35:22.
- TIMESTAMP: Similar a DATETIME, pero además puede actualizarse automáticamente al modificar un registro.
- YEAR: Almacena solo el año en formato de 4 dígitos. Ejemplo: 2025.



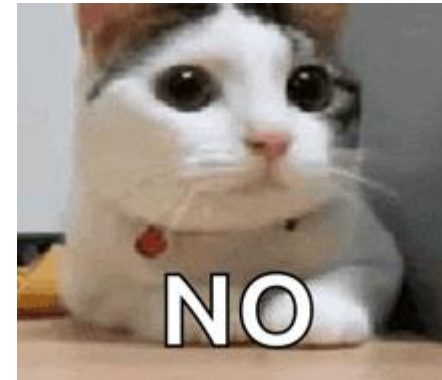
Data Type	Storage (bytes)	Date Range	Accuracy
DATETIME	8	January 1, 1753 to December 31, 9999	3-1/3 milliseconds
SMALLDATETIME	4	January 1, 1900 to June 6, 2079	1 minute
DATETIME2	6 to 8	January 1, 0001 to December 31, 9999	100 nanoseconds
DATE	3	January 1, 0001 to December 31, 9999	1 day
TIME	3 to 5		100 nanoseconds
DATETIMEOFFSET	8 to 10	January 1, 0001 to December 31, 9999	100 nanoseconds



# Tipos de datos booleanos

---

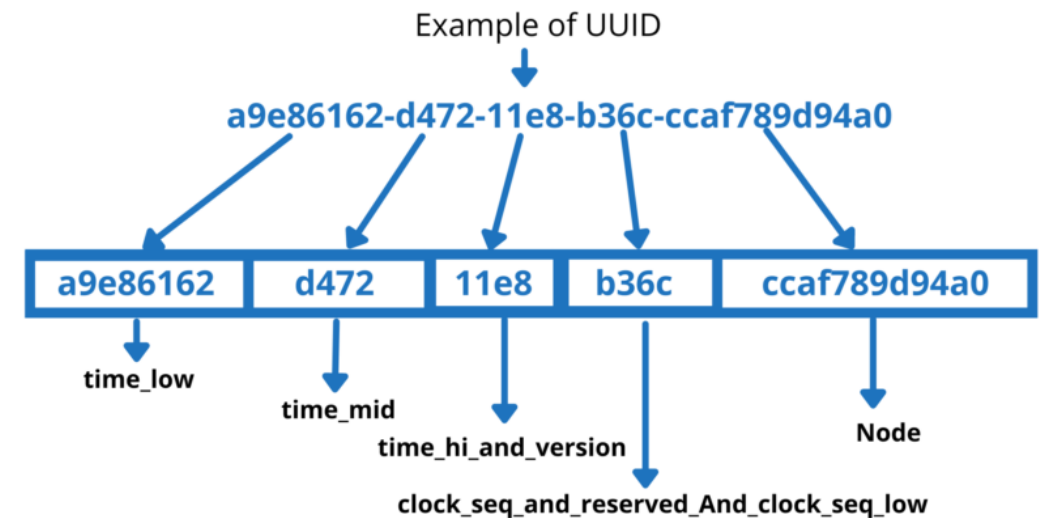
- BOOLEAN: Almacena valores de verdad (verdadero o falso)



# Tipos de datos de identificadores únicos

- UUID: Almacena identificadores únicos universales en formato UUID.

Los UUID son identificadores únicos universales que se utilizan para asegurar que los elementos en un sistema tengan identificadores que no se dupliquen, lo que es especialmente importante en sistemas distribuidos y bases de datos.





# Tipos de datos de colecciones y estructuras

---

- ARRAY: Almacena una colección de valores del mismo tipo en una matriz.
- JSON/JSONB: Almacena datos en formato JSON. JSONB ofrece un almacenamiento binario más eficiente y funciones de búsqueda.

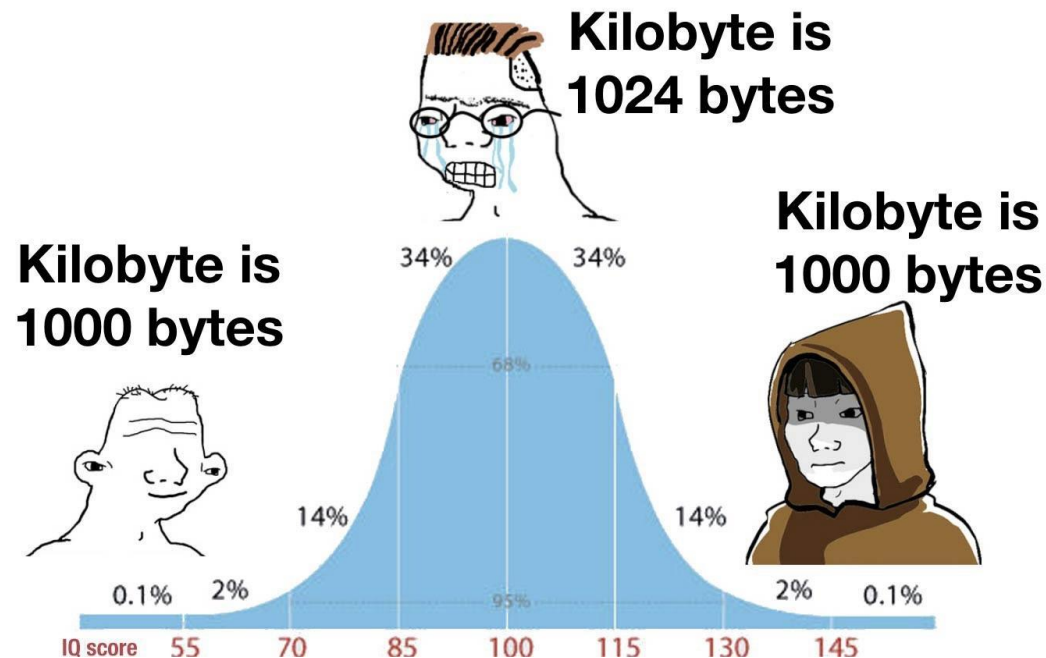


```
{
  "orders": [
    {
      "orderno": "748745375",
      "date": "June 30, 2088 1:54:23 AM",
      "trackingno": "TN0039291",
      "custid": "11045",
      "customer": [
        {
          "custid": "11045",
          "fname": "Sue",
          "lname": "Hatfield",
          "address": "1409 Silver Street",
          "city": "Ashland",
          "state": "NE",
          "zip": "68003"
        }
      ]
    }
  ]
}
```

# Tipos de datos binarios

---

- BYTEA: Almacena datos binarios, como imágenes o archivos.

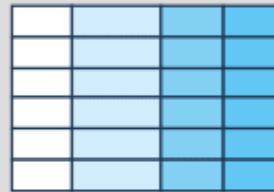


# Bases de Datos

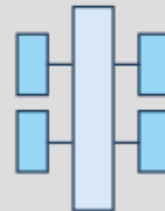
- ¿Cómo materializamos un modelo relacional en una DB?
- Existen 2 tipos de bases de datos. Relacionales y no relacionales.
- No hay una mejor que otra, siempre todo depende del contexto.
- En este curso se trabajará con bases de datos relacionales.

## SQL

### Relational

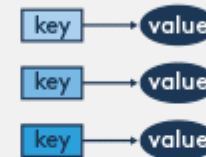


### Analytical (OLAP)



## NoSQL

### Key-Value



### Column-Family



### Graph



### Document

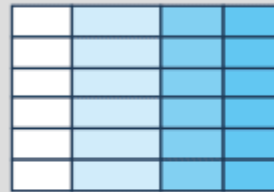


# SQL

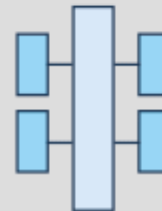
- SQL = Structured Query Language
- Lenguaje de programación.
- Diseñado para gestionar y manipular bdd relacionales.
- Fundamental para administración y análisis de datos de información estructurada

## SQL

### Relational



### Analytical (OLAP)



## NoSQL

### Key-Value



### Column-Family



### Graph



### Document

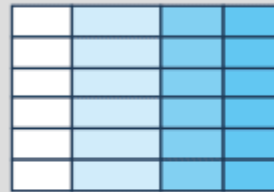


# SQL

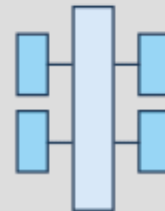
- Basado en tablas (Filas y columnas)
- Cada tabla posee un esquema fijo (estructura definida con tipos de datos)
- Las tablas se relacionan con PK y FK. Además con JOIN se puede combinar información.
- Sigue el modelo ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)

## SQL

### Relational

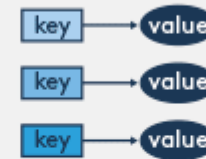


### Analytical (OLAP)



## NoSQL

### Key-Value



### Column-Family



### Graph



### Document



# Preparación entorno

---

```
CREATE DATABASE nombre_db;
```

```
CREATE TABLE nombre_tabla (  
  nombre_atributo1 tipo_atributo1,  
  nombre_atributo2 tipo_atributo2,  
  ...  
);
```

## SQL Worksheet

```
1  --MiguelTroyano.com|  
2  CREATE TABLE empleados (  
3      ID integer PRIMARY KEY,  
4      NOMBRE varchar(50),  
5      PUESTO varchar(50),  
6      SUELDO integer  
7  );
```

Table created.



# Entidad y Atributos

---

El comando SELECT se emplea para recuperar datos específicos de una o más tablas en una base de datos, donde debemos tener en cuenta tres cosas:

- Atributos: Atributos a mostrar
- Relación: Entidades de dónde se extraerán los datos
- Condición: Condiciones para filtrar los resultados Ejemplo: >, <, >=, <+, &&, ||

- **Atributos:** Atributos a mostrar (**SELECT**)
- **Relación:** Entidades de dónde se extraerán los datos (**FROM**)
- **Condición:** Condiciones para filtrar los resultados (**WHERE**)  
Ejemplo: >, <, >=, <+, &&, ||

```
SELECT atributo(s)  
FROM tabla(s)  
WHERE condicion(es)
```

# Ejercicio 1

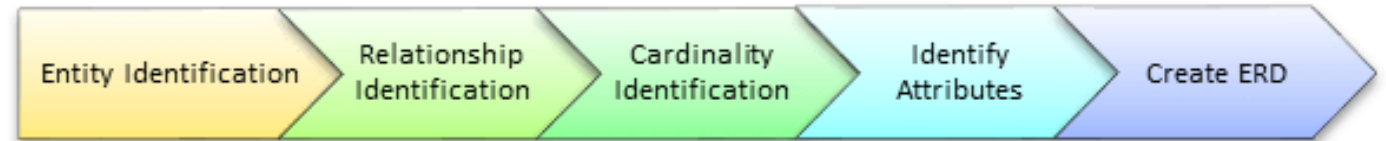
---

```
SELECT atributo(s)
FROM tabla(s)
WHERE condicion(es)
```

Buscar empleados con sueldos sobre 8000 y que trabajan en la empresa desde el primero de enero del 2003.

**Empleado**

RFC	Nombre	sueldo	depto	ingreso
SACV750524	Verónica Sánchez	\$ 6,000.00	A1	01/01/2004
HEFC790926	Carlos Hernández	\$ 8,000.00	A2	01/01/2003
ROPJ081001	Jorge Rojas	\$ 10,000.00	A2	01/10/2003
ROAO740114	Oscar Romero	\$ 8,000.00	A1	01/03/2002
FORG760206	Germán Flores	\$ 12,000.00	A1	01/01/2002
PEVN780704	Noé Pérez	\$ 15,000.00	A3	01/06/2002
JUBF910410	Fabián Juárez	\$ 6,000.00	A2	01/11/2003
HEVC950826	Carlos Hernández	\$ 12,000.00	A1	01/06/2002
RUZP810512	Patricia Ruíz	\$ 8,000.00	A2	01/04/2003
LOMM750919	María López	\$ 10,000.00	A3	01/10/2002



## Ejercicio 2

---

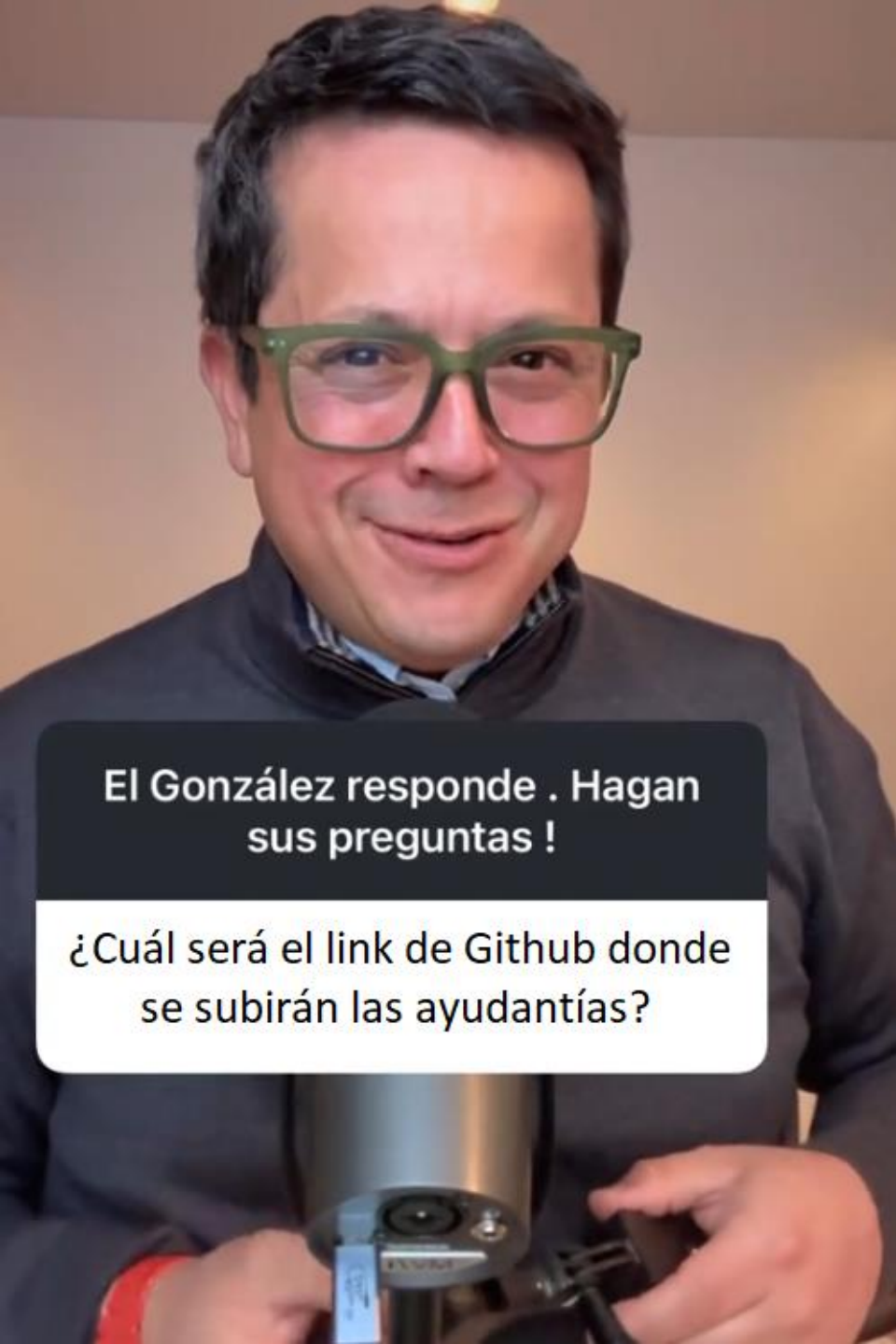
En un complejo turístico se busca implementar un sistema para administrar el uso y arriendo de sus cabañas. Cada cabaña se identifica y cuenta con información relacionada a su capacidad, comodidades y valor diario de arrendamiento. Los huéspedes que llegan al complejo deben registrarse con sus datos personales para poder realizar arriendos, los cuales pueden abarcar una o varias cabañas durante un cierto período de tiempo. Un mismo huésped puede arrendar diferentes cabañas en distintos momentos, y a su vez, cada cabaña puede recibir a diversos huéspedes a lo largo del tiempo. En la gestión del lugar también intervienen los empleados, quienes tienen funciones específicas dentro del complejo y pueden estar a cargo de una o varias cabañas, así como cada cabaña puede ser atendida por distintos empleados. Los arriendos quedan registrados con la información necesaria sobre las fechas, el monto a cancelar y el huésped que los realiza, de modo que se pueda llevar un control de las estadías y de la ocupación de las cabañas.



## Ejercicio 2

---

- 1) Obtener el nombre y teléfono de los huéspedes que viven en Santiago.
- 2) Listar las cabañas con capacidad mayor a 5 personas.
- 3) Mostrar las cabañas cuya tarifa diaria sea menor a 50.000.
- 4) Obtener los empleados que trabajan como recepcionistas.
- 5) Listar los arriendos realizados después del 1 de enero de 2025.
- 6) Obtener las cabañas que tienen más de 2 baños.



El González responde . Hagan  
sus preguntas !

¿Cuál será el link de Github donde  
se subirán las ayudantías?

# Link Github

---

Repositorio:

<https://github.com/VicenteDiazH/AY-BDD2025-1-S1>