

# Proyecto DB

## Películas

### PALOMITAS

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €

### BEBIDAS

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €

### COMBO

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €



# Operadores

## Operadores relacionales

- \$eq ➡ Igual
- \$ne ➡ Diferente
- \$gt ➡ Mayor que
- \$gte ➡ Mayor que (estricto)
- \$lt ➡ Menor que (estricto)
- \$lte ➡ Menor que
- \$in ➡ Contiene
- \$nin ➡ No contiene

## Operadores relacionales

- \$and ➡ Cumple alguna de las condiciones
- \$or ➡ Cumple todas las condiciones

# Map-Reduce

## Map-Reduce

*Map-reduce es un paradigma de procesamiento de datos para condensar grandes volúmenes de datos en resultados agregados útiles*

- **Map Reduce** permite realizar acciones sobre datos de forma paralela y altamente escalable
- Es un modelo de programación que consta de dos pasos: Map y Reduce
- **Map:** La información completa se separa en bloques, cada uno con una clave. Se realiza alguna transformación y se envía cada bloque al paso de Reduce
- **Reduce:** Un Reduce toma los datos que salen del paso Map (que tengan la misma clave) y los procesa

Las agregaciones son más rápidas porque usan c / c++ (como el código core de mongo) y los map-reduce usan JSON, por lo que tienen que convertir cada archivo interno BSON a JSON para poder operar con ellos

PALOMITAS



# Aggregate

## Aggregate

*Las agregaciones procesan varios documentos en pasos correlativos. Estos son los 5 steps más comunes:*

- **\$group**: Agrupa documentos y añade cuantificadores
- **\$match**: Filtrado de documentos igual que las queries
- **\$lookup**: Busca en otras colecciones
- **\$project**: Deja, o no, visible un set de atributos
- **\$unwind**: Desmonta Arrays

## Index

*Los índices admiten la ejecución eficiente de consultas. Sin índices, MongoDB debe realizar un escaneo de la colección completa, es decir, escanear cada documento en una colección, para seleccionar aquellos documentos que coincidan con la declaración de la consulta*

PALOMITAS

{

**\_id**: Identificador único MongoDB,

**adult**: Adultos si/no,

**belongs to collection**: {

**id**: Id de colección,

**name**: Nombre colección

},

**budget**: Presupuesto,

**genres**:[

{

**id**: Identificador de género,

**name**: Nombre del género,

}

],

**id**: Identificador unitivo,

**original\_language**: Abreviatura del idioma,

**original\_title**: Nombre original,

**popularity**: Popularidad,

**production\_companies**:[

{

**name**: Nombre compañía,

**id**: Identificador compañía,

}

],

**revenue**: Ingresos,

**runtime**: Duración min,

**spoken\_languages**:[

{

**iso\_639\_1**: Identificador idiomas,

**name**: Nombre idioma,

}

],

**status**: Estado, lanzada o no,

**tagline**: Marketing,

**title**: Título,

**vote\_average**: Promedio de votos,

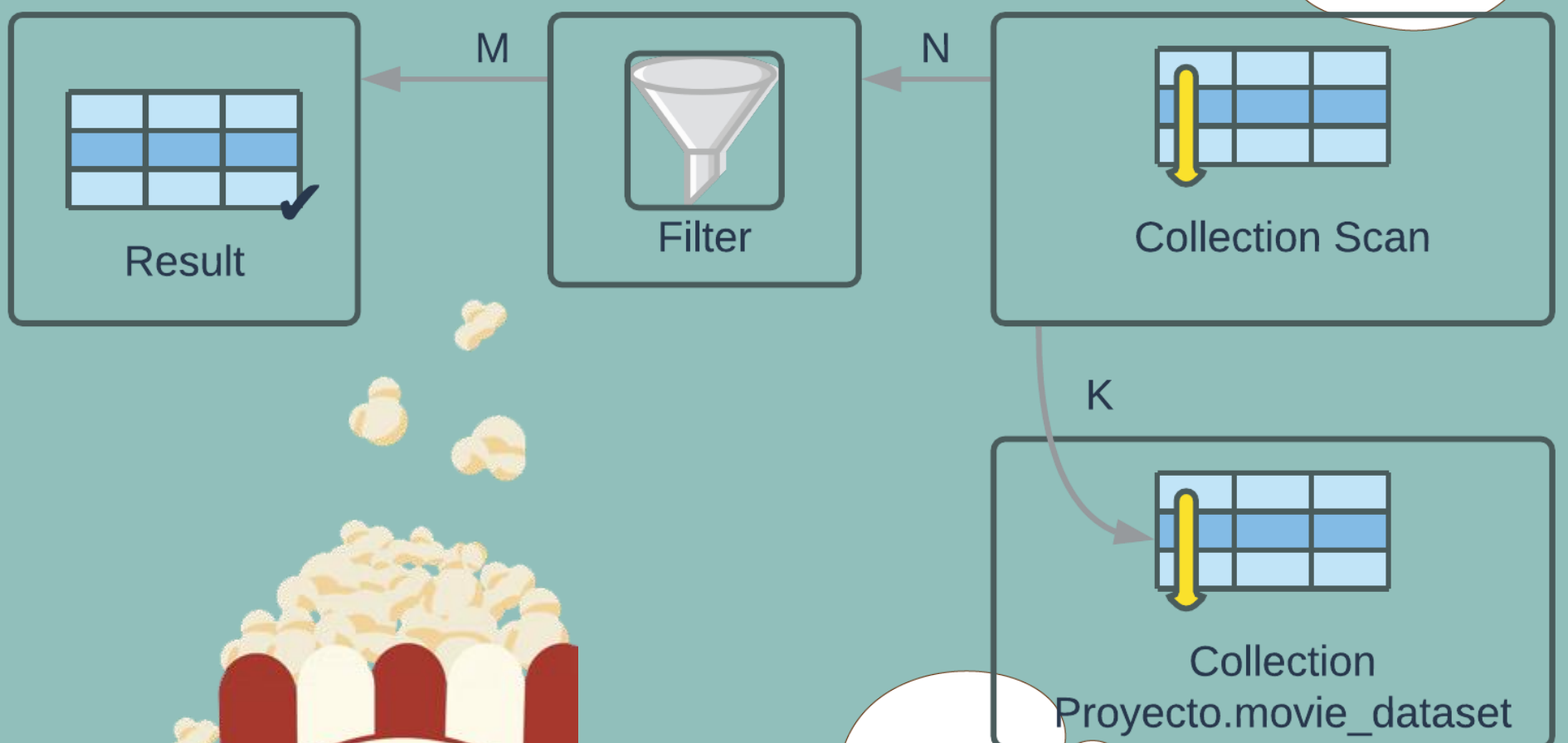
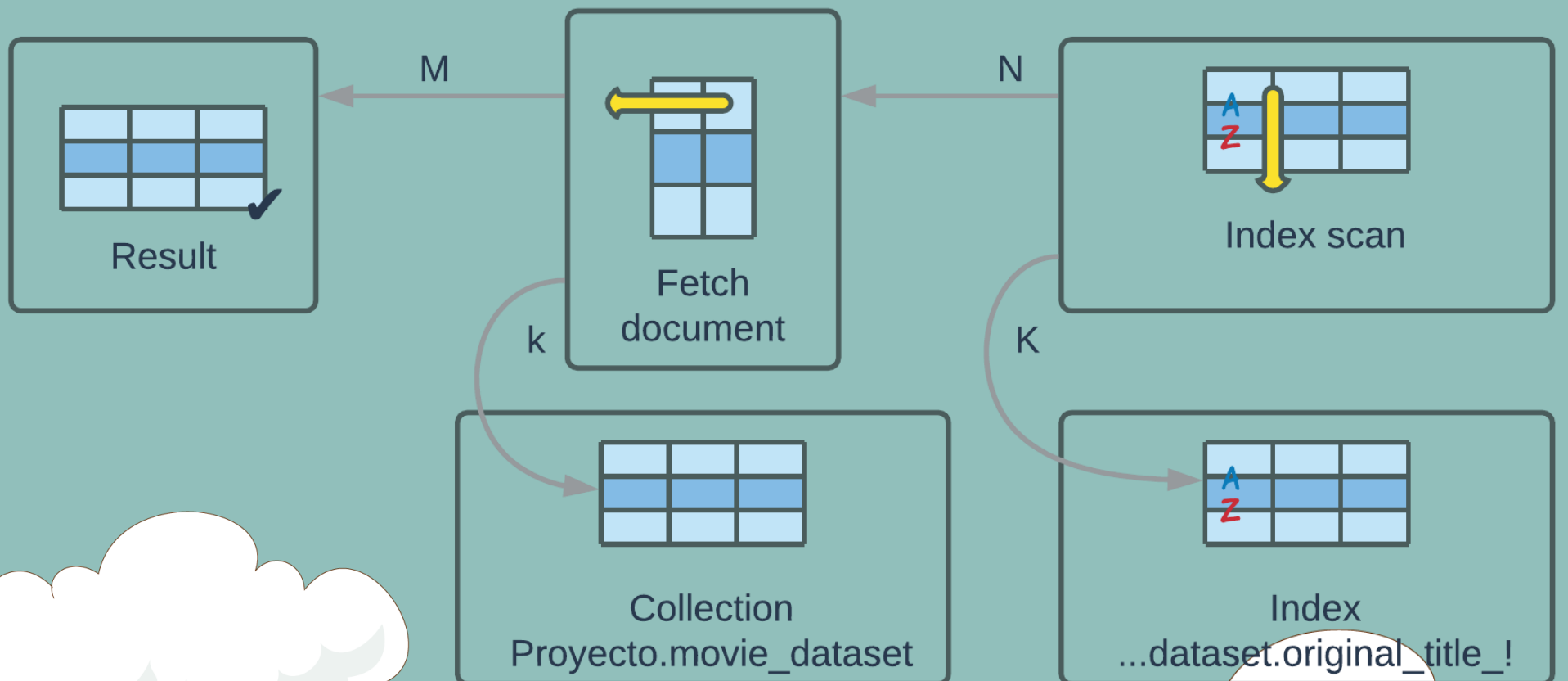
**vote\_count**: Número de votos,

**releasedate**: Fecha lanzamiento

}

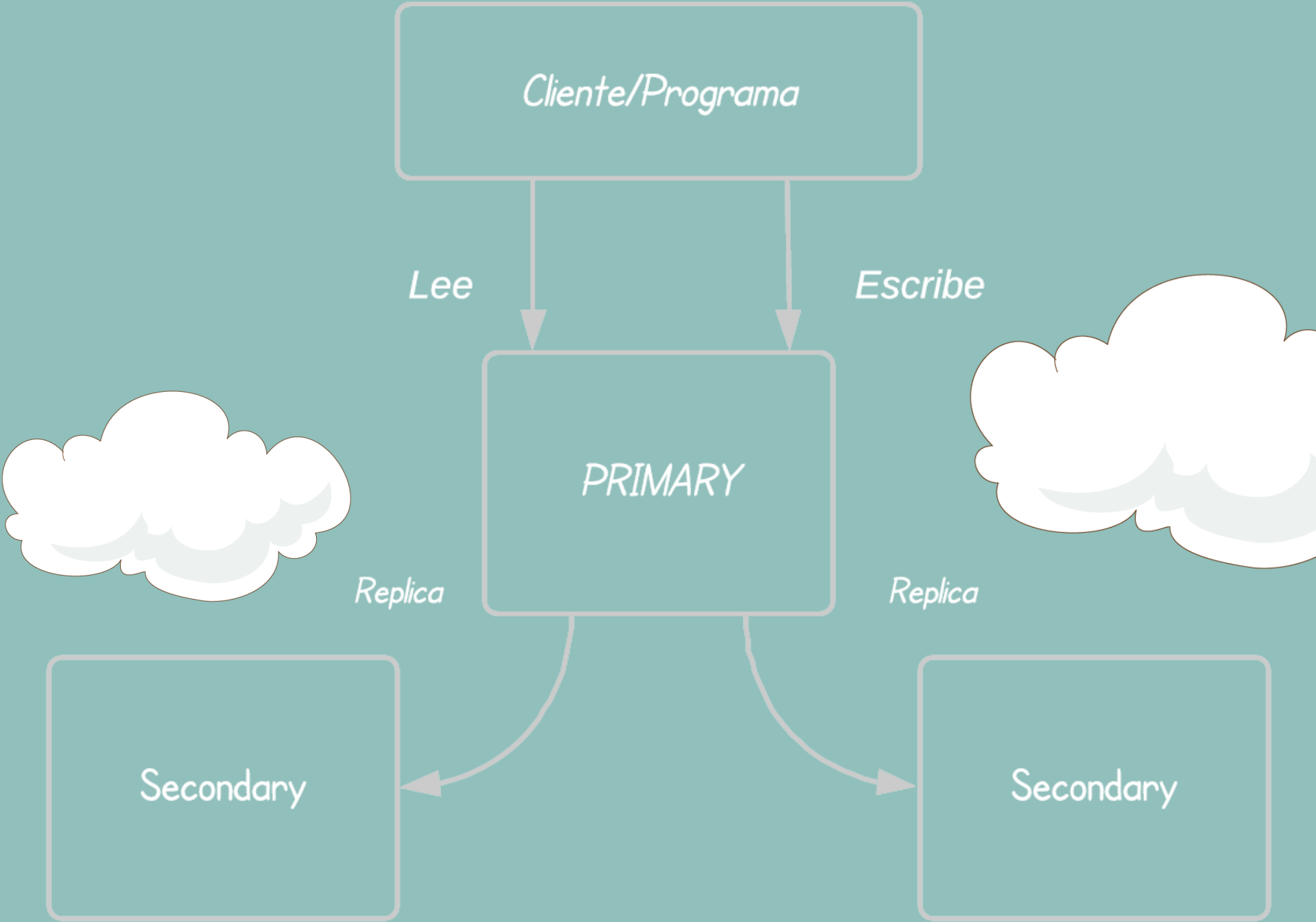


# Explain



PALOMITAS

# ReplicaSet





# Sharding

## Sharding

*El sharding es un método para la distribución de datos en varias máquinas. MongoDB utiliza el sharding para admitir implementaciones con conjuntos de datos muy grandes y operaciones de alto rendimiento.*

- **Distribución** de los documentos de una colección en diferentes servidores
- Cada shard puede ser o bien una **instancia** de mongo o un **replicaset**
- Búsqueda de una buena **sharding key** -> como las funciones de hash para repartir bien los datos

PALOMITAS



FIN

## PALOMITAS

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €

## BEBIDAS

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €

## COMBO

Pequeño .....	5.00 €
Mediana .....	10.00 €
Grande .....	15.00 €

