

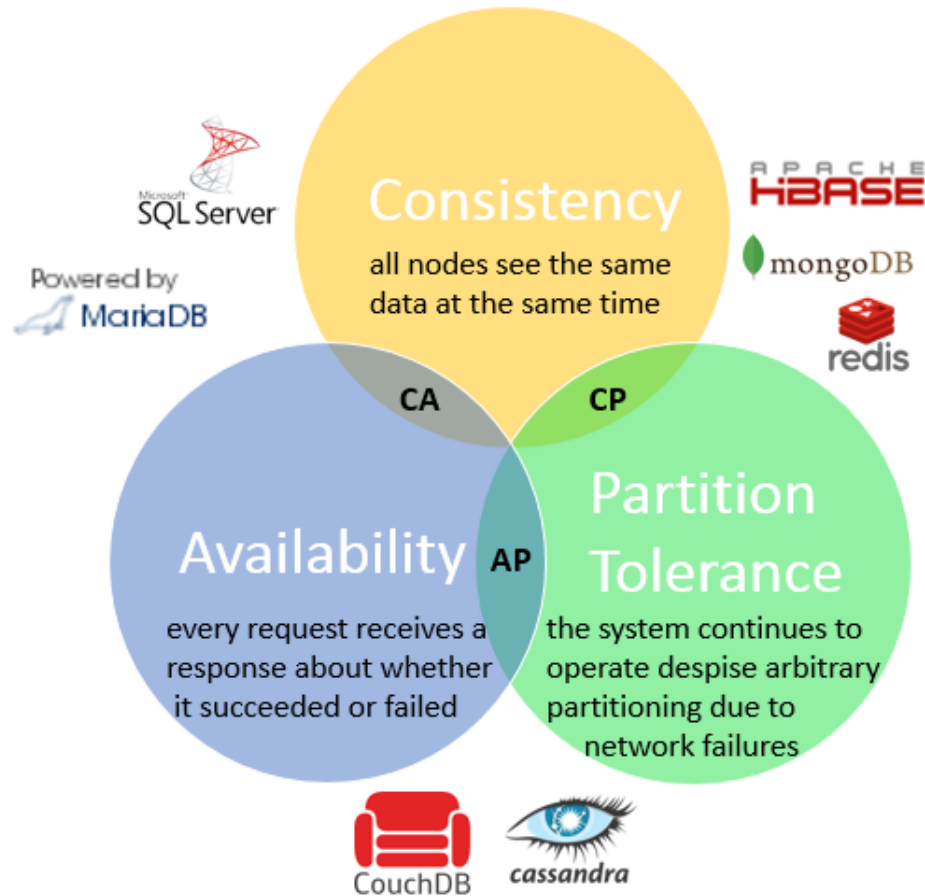
# **CIERRE TÉCNICAS AVANZADAS DE BASE DE DATOS**

**SERGIO ÁLVAREZ**

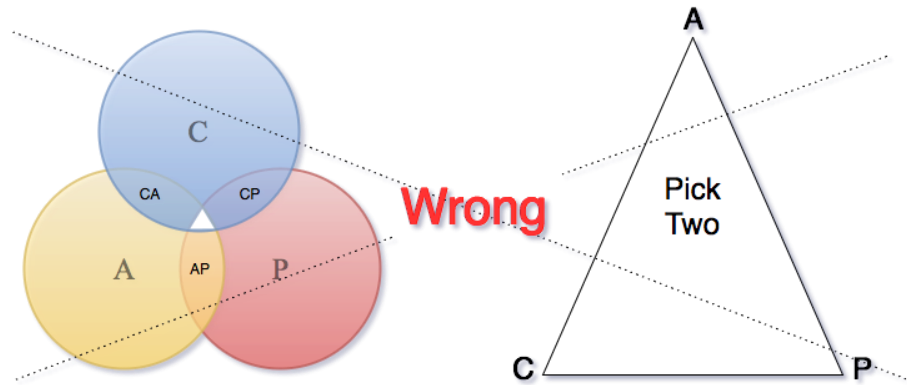
**VERSIÓN 1.2**

# CONCEPTOS BÁSICOS

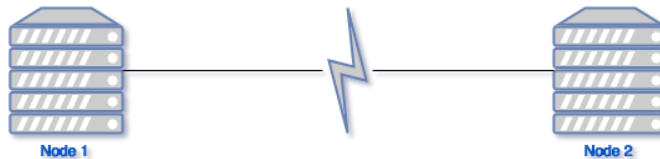
# TEOREMA DE CAP



# ANTI TEOREMA DE CAP



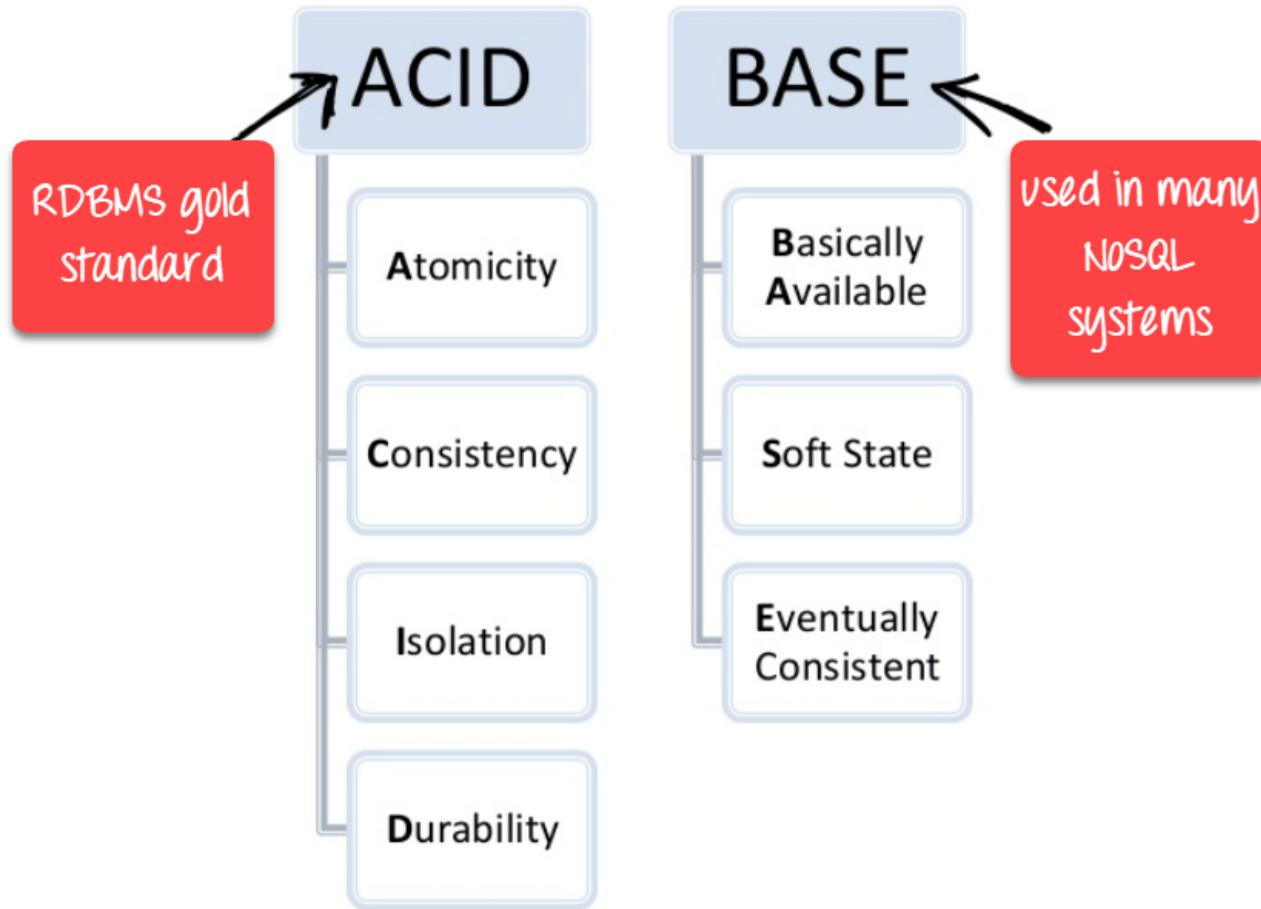
In the event of a partition choose one



vs



# ACID VS BASE



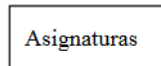
# PROCESO DISEÑO



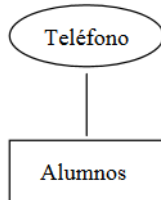
# MODELO LÓGICO

# MODELO ENTIDAD RELACIÓN

- Tipos de entidades: Rectángulos.



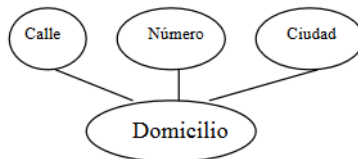
- Atributos: Elipses. Se conectan mediante líneas a los tipos de entidades o tipos de relación.



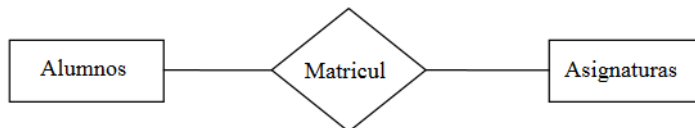
- Atributos multivalorados: Una elipse con doble línea:



- Atributos compuestos. Los componentes de un atributo se representan a su vez como atributos:

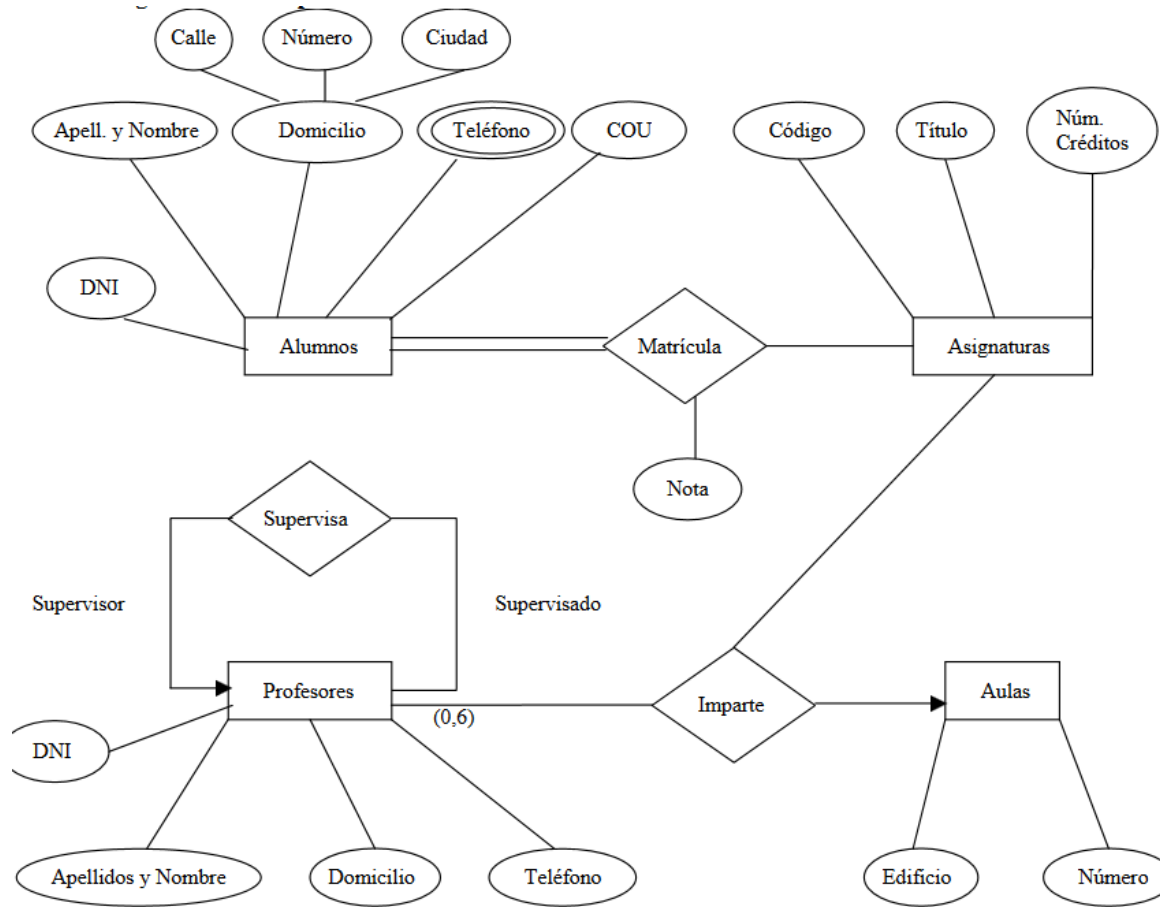


- Tipos de Relación: Rombos conectados a los tipos de entidades que relacionan.

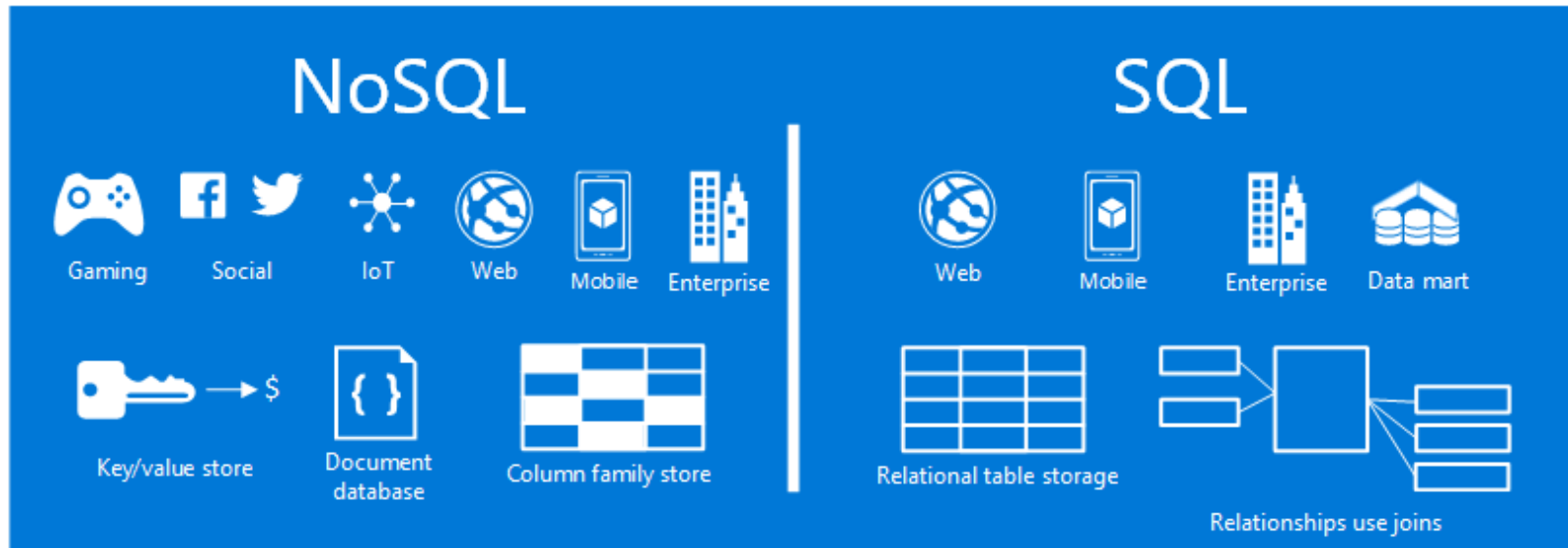




# MODELO ENTIDAD RELACIÓN

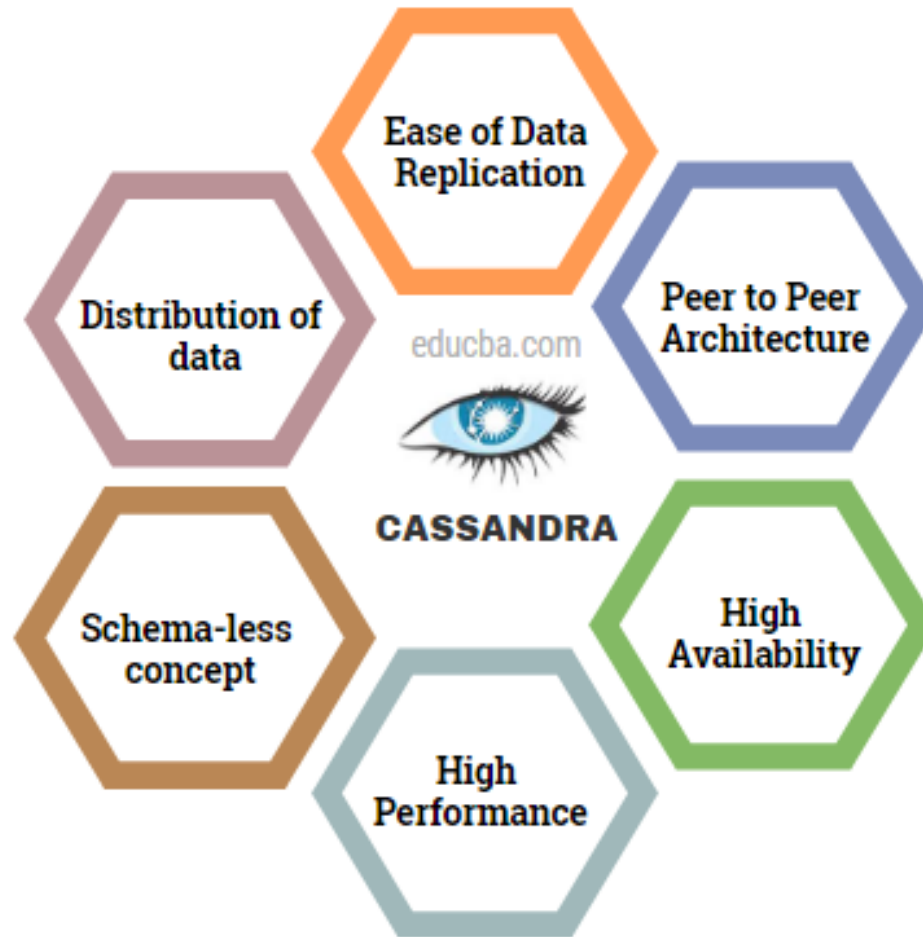


# NOSQL VS SQL

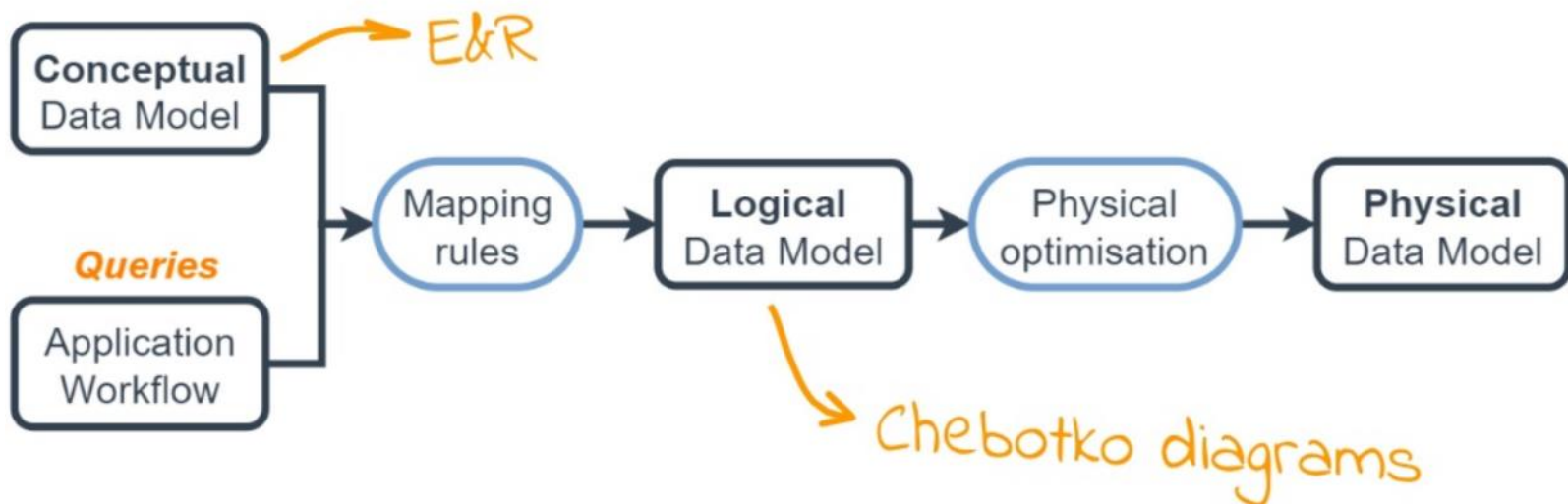


# NOSQL – APACHE CASSANDRA

# QUE ES CASSANDRA

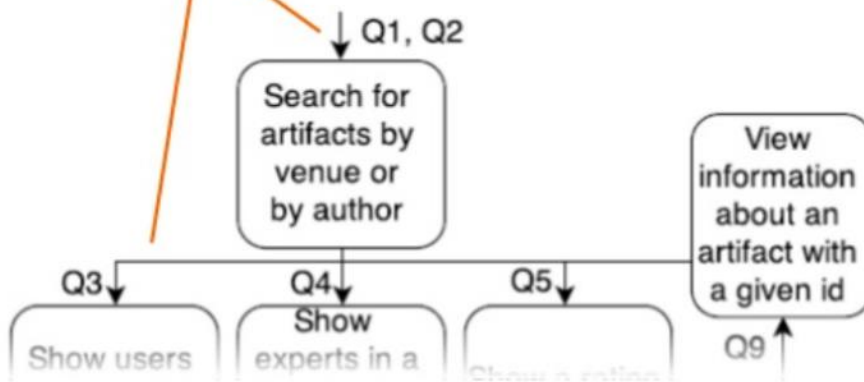


# APACHE CASSANDRA



# APACHE CASSANDRA

Query workflow



Query list

Q1: Find artifacts published in a venue with a given name after a given year. Order results by year (DESC).

Q2: Find artifacts published by a given author. Order results by year (DESC).

Q3: Find users who liked a given artifact.

Q4: Find users who liked a given artifact and who have expertise in a certain area.

Q5: Find average rating of artifact with a given id.

# APACHE CASSANDRA

Table Name			
column name 1	<i>CQL-Type</i>	K	← Partition key column
column name 2	<i>CQL-Type</i>	C↑	← Clustering key column (ASC)
column name 3	<i>CQL-Type</i>	C↓	← Clustering key column (DESC)
column name 4	<i>CQL-Type</i>	S	← Static column
column name 5	<i>CQL-Type</i>	IDX	← Secondary index column
column name 6	<i>CQL-Type</i>	++	← Counter column
[column name 7]	<i>CQL-Type</i>		← Collection column (list)
{column name 8}	<i>CQL-Type</i>		← Collection column (set)
<column name 9>	<i>CQL-Type</i>		← Collection column (map)
column name 10	<i>CQL-Type</i>		← Regular column

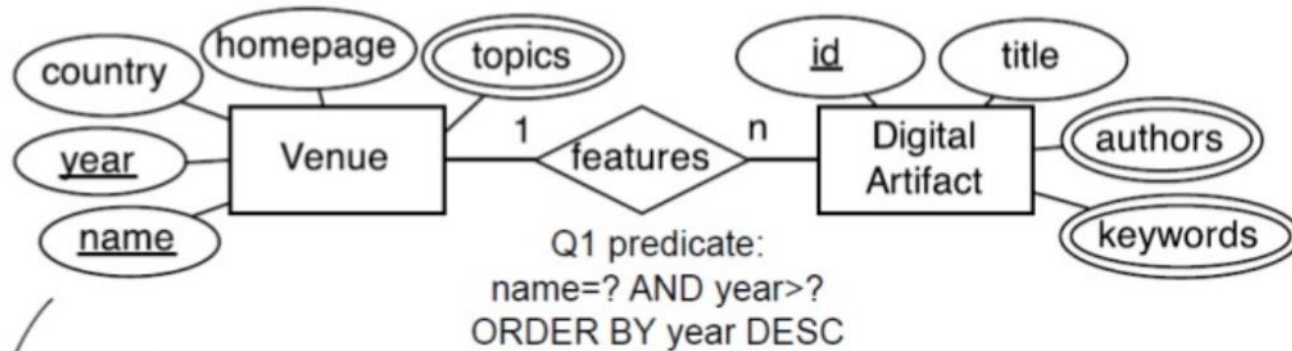
# APACHE CASSANDRA

```
CREATE TABLE actors_by_video (  
  video_id uuid,  
  actor_name text,  
  character_name text,  
  PRIMARY KEY ((video_id), actor_name, character_name)  
);
```

actors_by_video		
video_id	uuid	K
actor_name	text	C↑
character_name	text	C↑



# APACHE CASSANDRA



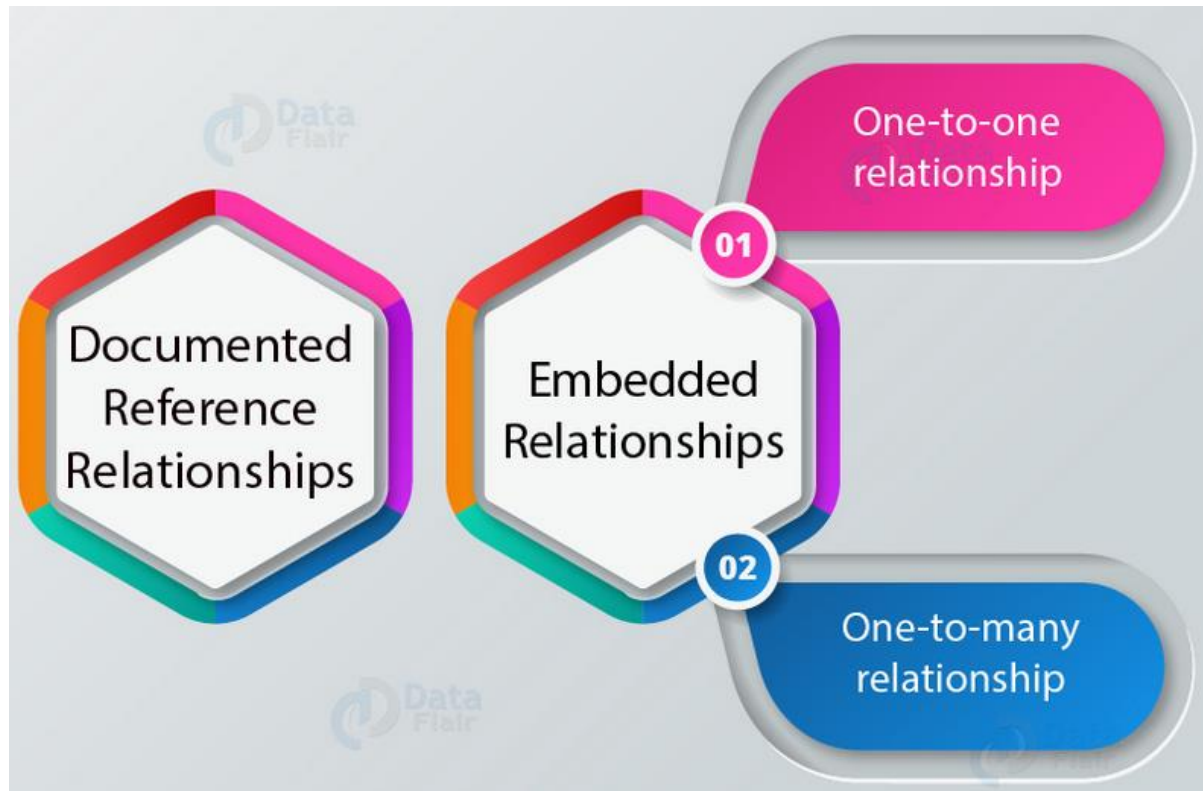
MR1	MR2	MR3	MR4	MR5
Artifacts_by_venue	Artifacts_by_v..	Artifacts_by_v..	Artifacts_by_v..	Artifacts_by_venue
venue_name	venue_name K	venue_name K	venue_name K	venue_name K
year	year	year C↑	year C↓	year C↓
artifact_id	artifact_id	artifact_id	artifact_id	artifact_id C↑
artifact_title	artifact_title	artifact_title	artifact_title	artifact_title
[authors]	[authors]	[authors]	[authors]	[authors]
{keywords}	{keywords}	{keywords}	{keywords}	{keywords}

# NOSQL – MONGO

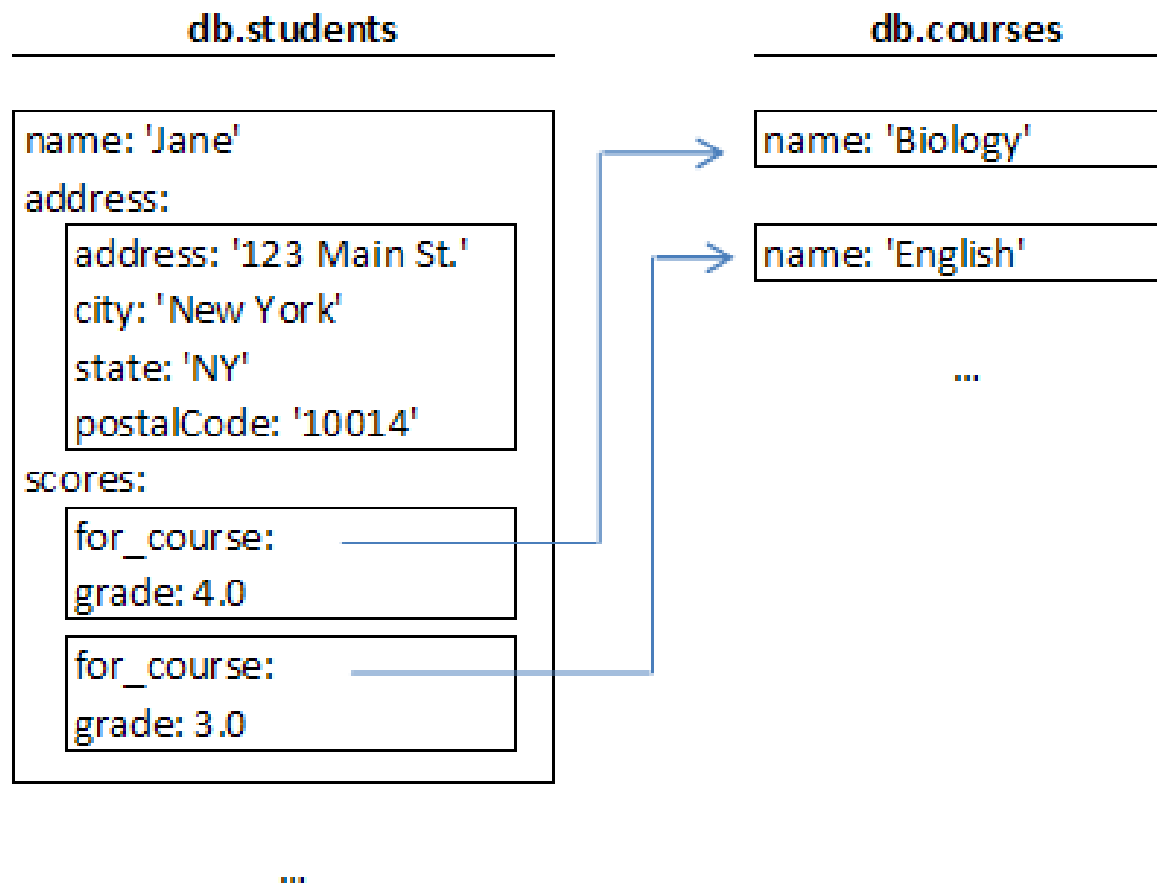
# QUE ES MONGO



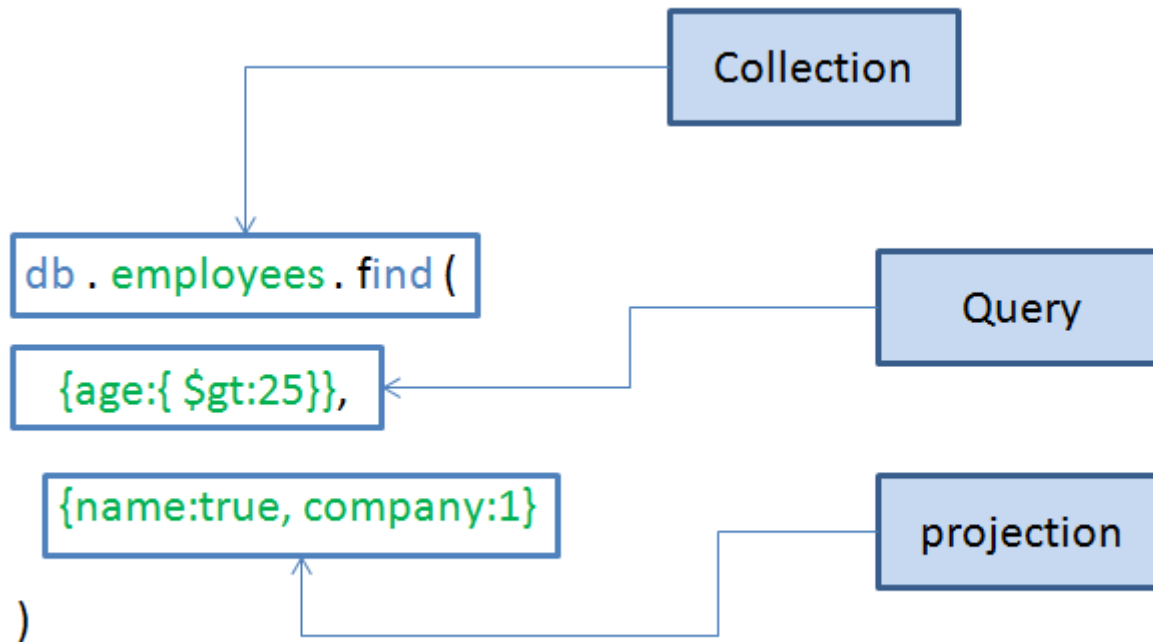
# MONGODB RELATIONSHIPS (EMBEDDED & REFERENCE)



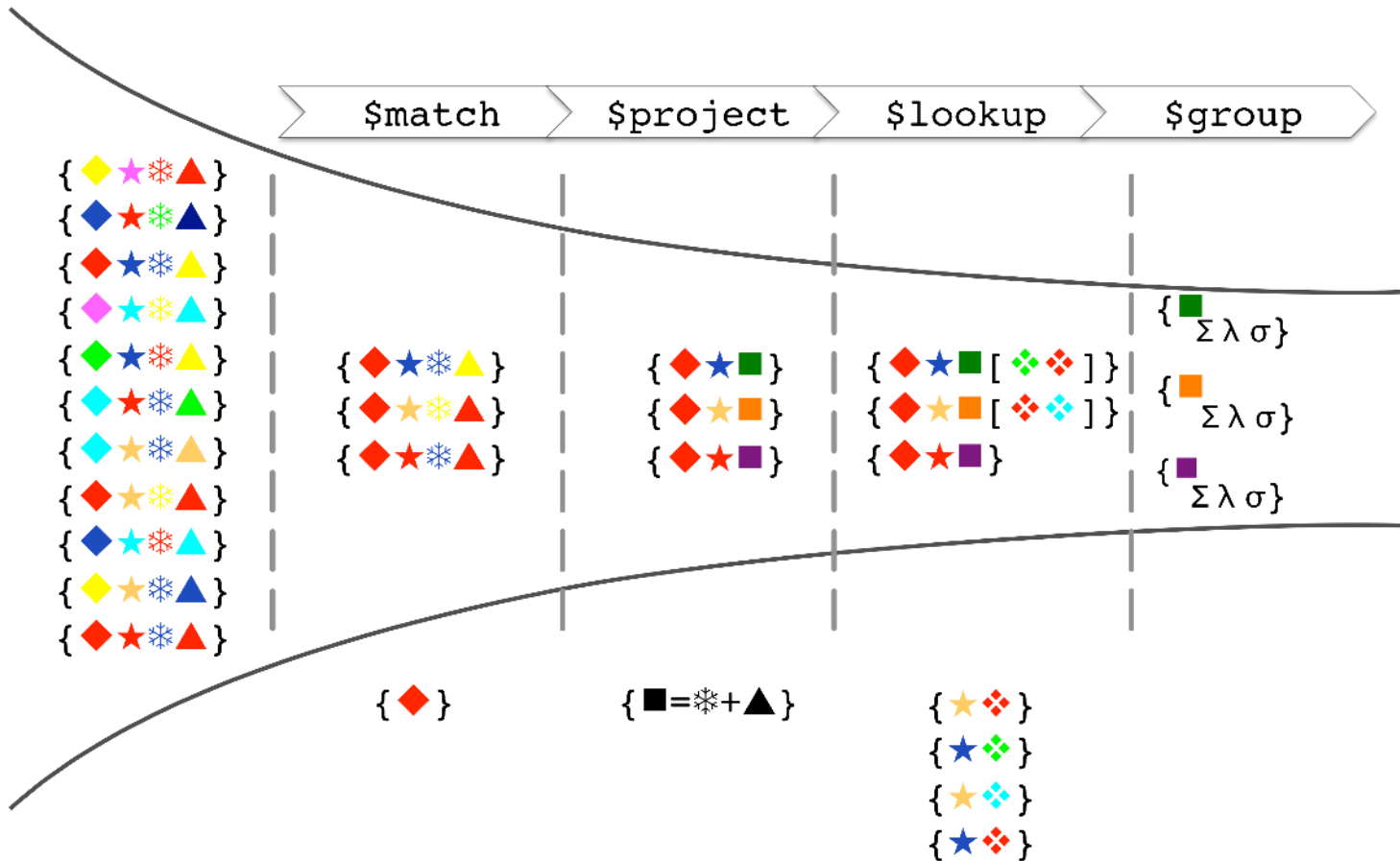
# MONGODB RELATIONSHIPS (EMBEDDED & REFERENCE)



# CONSULTAS



# CONSULTAS AVANZADAS



# CONSULTAS AVANZADAS

|||

\$match

☒

+

```
1 /**
2  * query - The query in MQL.
3  */
4 {
5   time_stamp: {$gte: ISODate("2018-08-13 17:33:20.
6  }
```

Output after \$match stage (Sample of 20 documents)

```
_id: "f2b2880f-3f56-4ec6-b11c-012ec29d3c22"
time_stamp: 2018-08-13 17:35:12.000
info_spot: null
query: "note de frais"
purpose: "SearchResults"
view: "Search"
number_of_results: 62
response_time: 176
```

```
_id: "a4cle4cb-d447-40fd-8fb8-c135abd99713"
time_stamp: 2018-08-13 17:35:54.000
info_spot: null
query: "{uid_search:*management* OR
givenname_search:*management* OR sn_search.
purpose: "Talent"
view: "NO_VIEW_USED"
number_of_results: 81
response_time: 260
```

|||

\$group

☒

+

```
1 /**
2  * id - The id of the group.
3  * field1 - The first field name.
4  */
5 {
6   _id: "$purpose",
7   groupedPurpose: {
8     $push: "$purpose"
9   }
10 }
```

Output after \$group stage (Sample of 4 documents)

```
_id: "PlateformeSportifsAPI"
groupedPurpose: Array
  0: "PlateformeSportifsAPI"
  1: "PlateformeSportifsAPI"
```

```
_id: "plateformsportif"
groupedPurpose: Array
  0: "plateformsportif"
  1: "plateformsportif"
  2: "plateformsportif"
  3: "plateformsportif"
  4: "plateformsportif"
  5: "plateformsportif"
  6: "plateformsportif"
```

```
_id: "plateformsportif"
groupedPurpose: Array
  0: "plateformsportif"
  1: "plateformsportif"
  2: "plateformsportif"
  3: "plateformsportif"
  4: "plateformsportif"
  5: "plateformsportif"
  6: "plateformsportif"
```

|||

\$project

☒

+

```
1 /**
2  * specifications - The fields to
3  * include or exclude.
4  */
5 {
6   number_of_results: {$size: "$groupedPurpose"}
7 }
```

Output after \$project stage (Sample of 4 documents)

```
_id: "PlateformeSportifsAPI"
number_of_results: 2
```

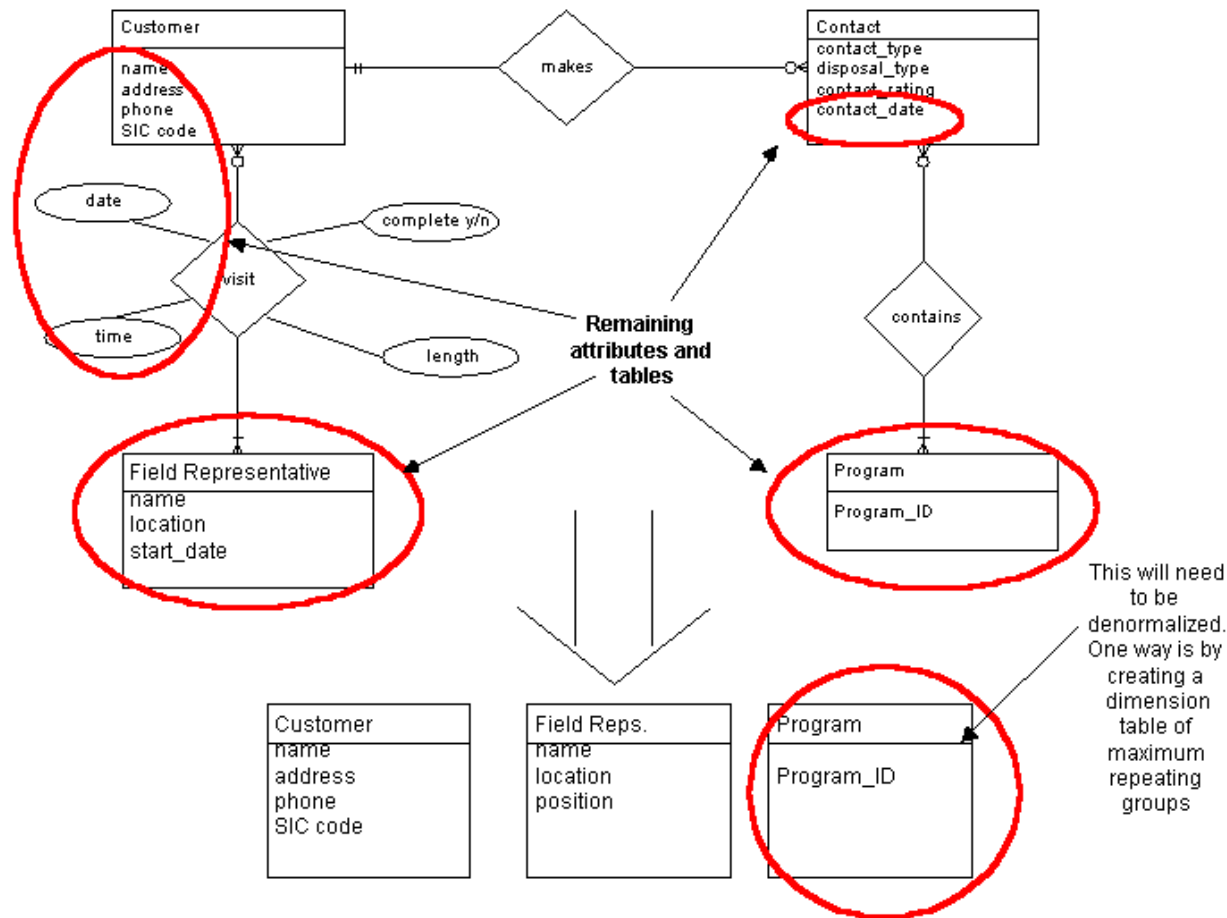
```
_id: "plateformsportif"
number_of_results: 26
```

```
_id: "plateformsportif"
number_of_results: 26
```

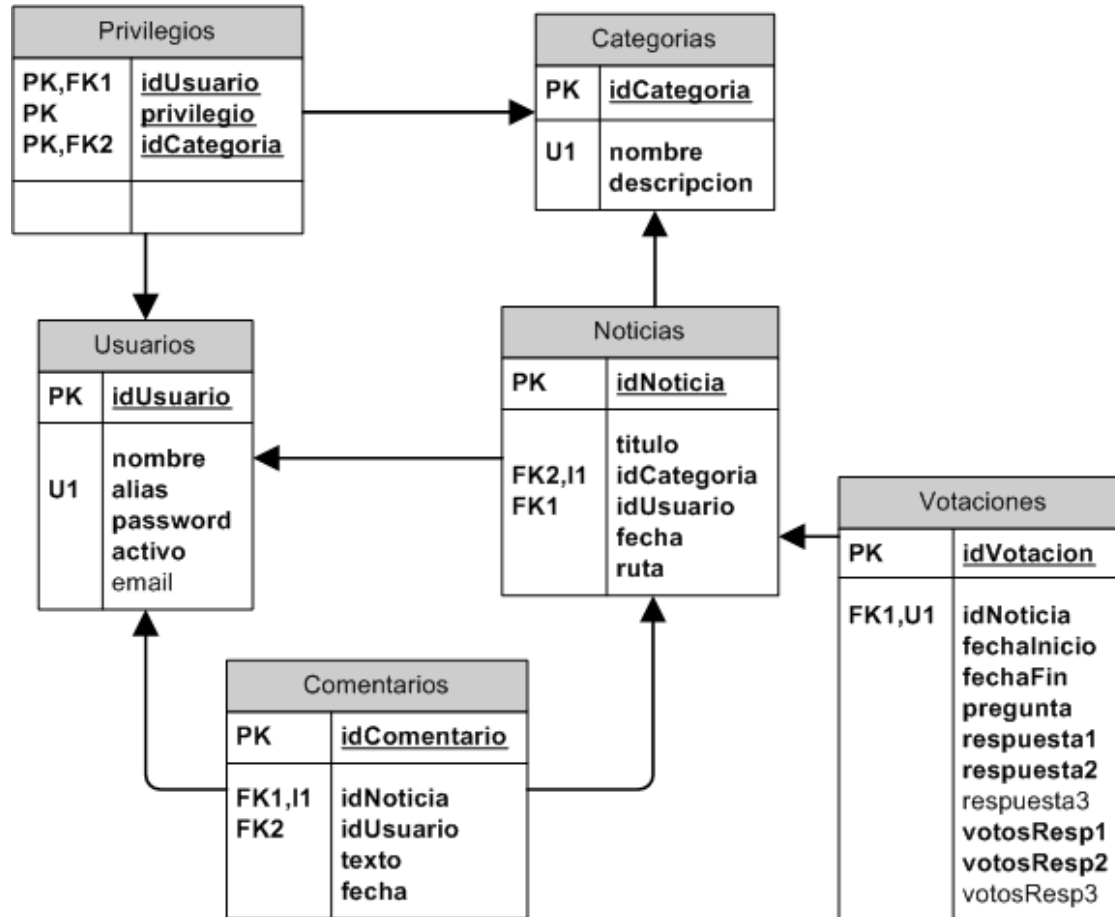


# **BASES DE DATOS RELACIONALES**

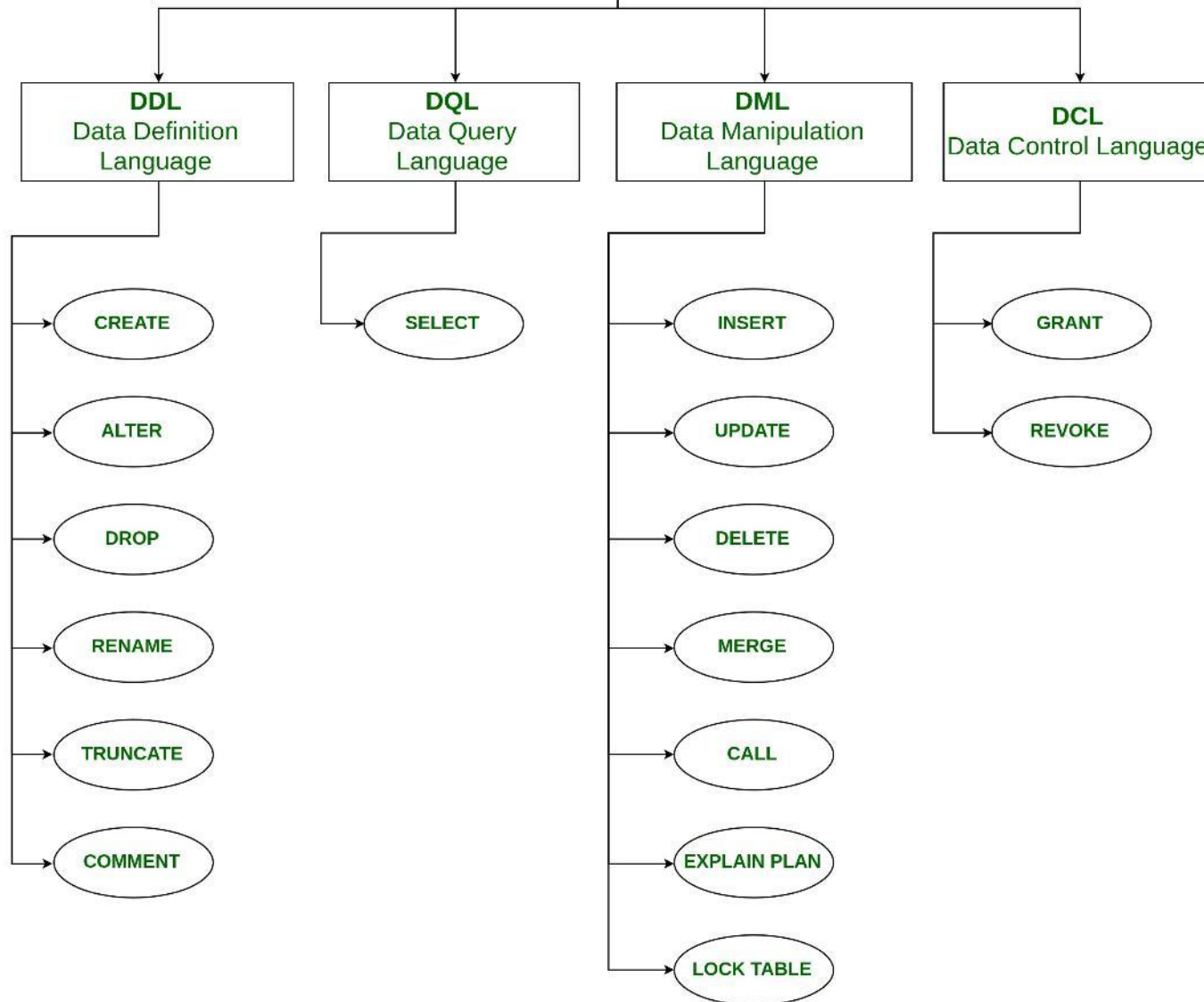
# MODELO E-R A RELACIONAL



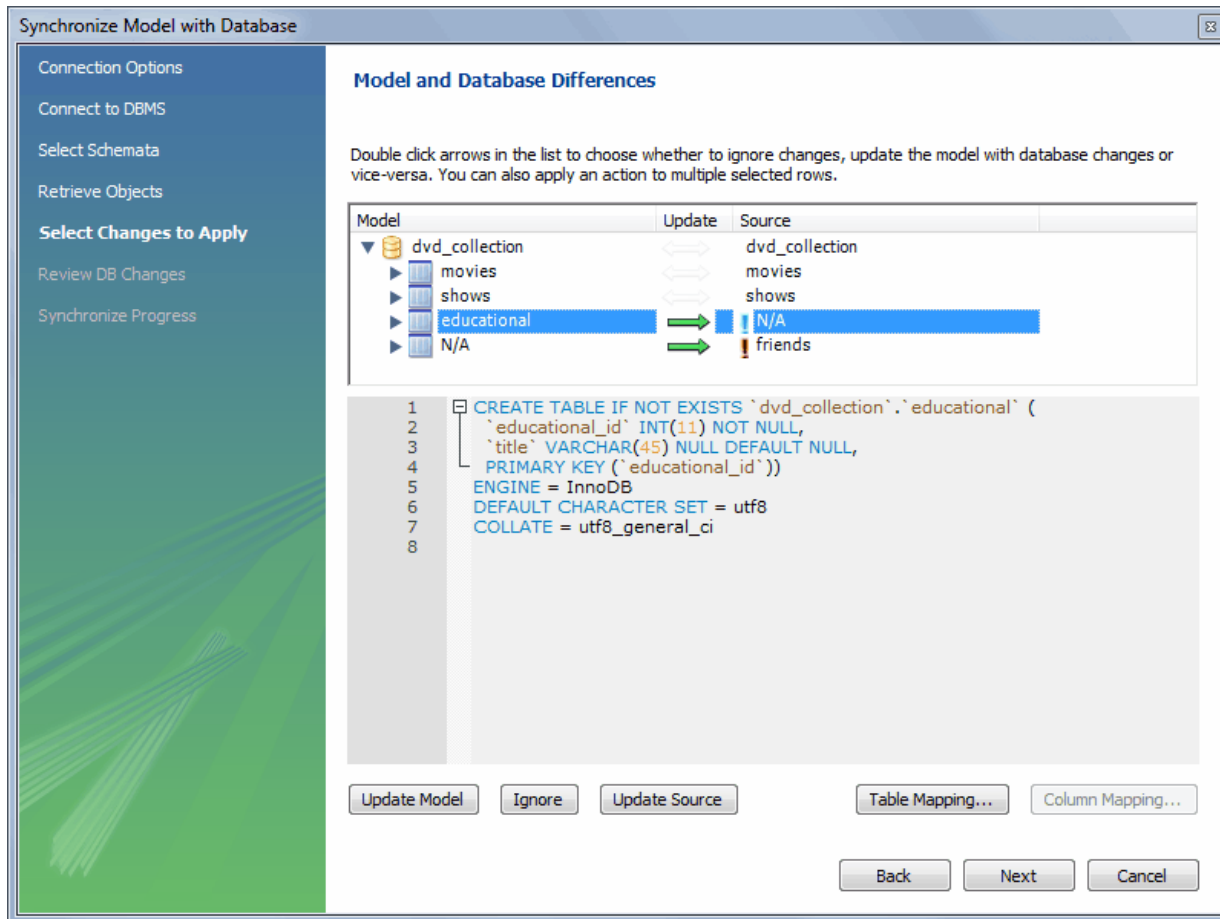
# MODELO RELACIONAL



# COMANDOS SQL



# HERRAMIENTAS CASE



# OPCIÓN PROCEDIMENTAL

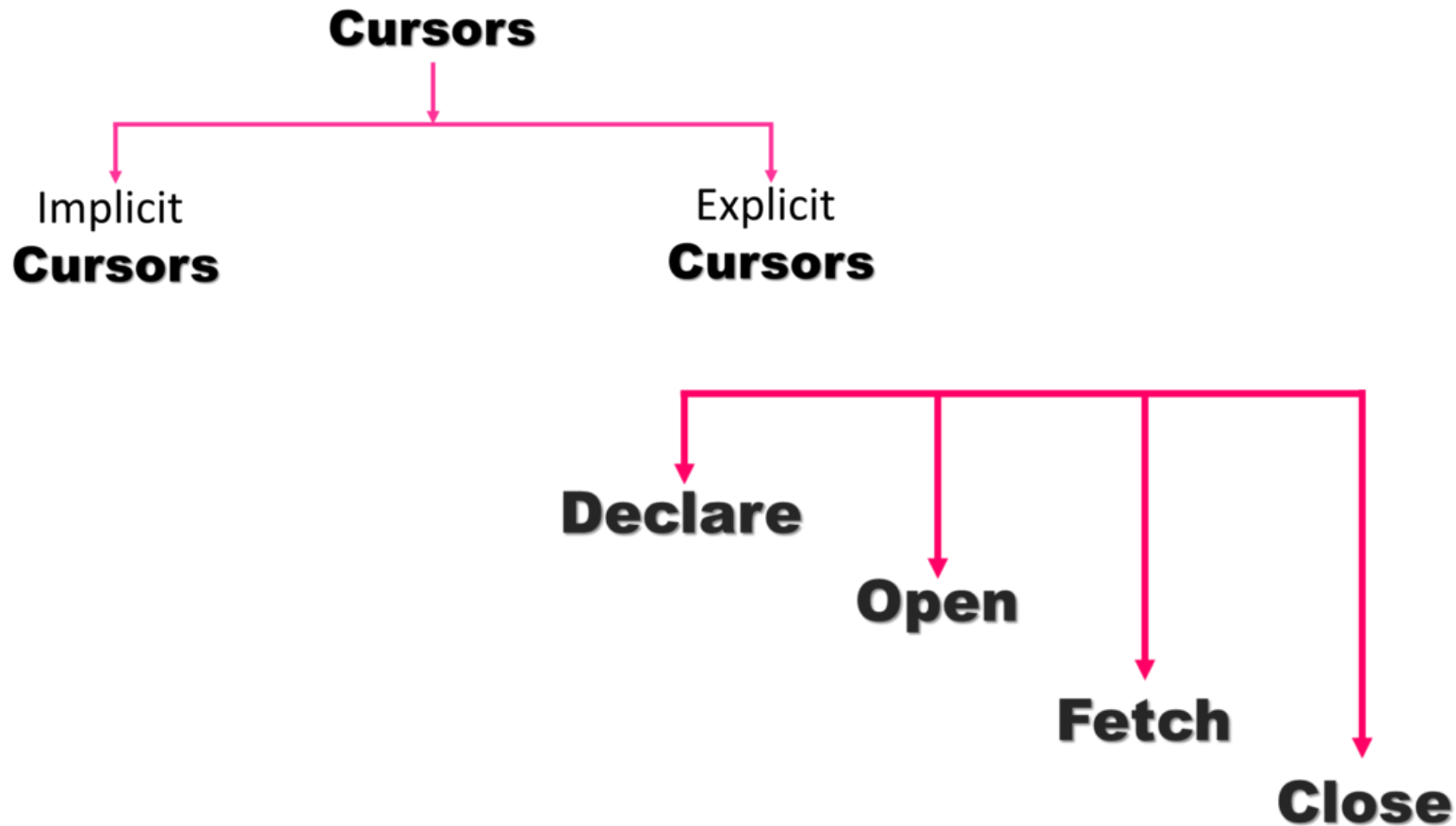
```
Query sp2 x +
1 DELIMITER $$
2
3 CREATE
4     /*[DEFINER = { user | CURRENT_USER }]*/
5     PROCEDURE `world`.`sp2`()
6     /*LANGUAGE SQL
7      | [NOT] DETERMINISTIC
8      | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
9      | SQL SECURITY { DEFINER | INVOKER }
10     | COMMENT 'string'*/
11     BEGIN
12     INSERT INTO `world`.`countrylanguage` (`CountryCode`) VALUES ('088');
13     END$$
14
15 DELIMITER ;
```

# **TEMAS NUEVOS**

## **BASES DE DATOS**

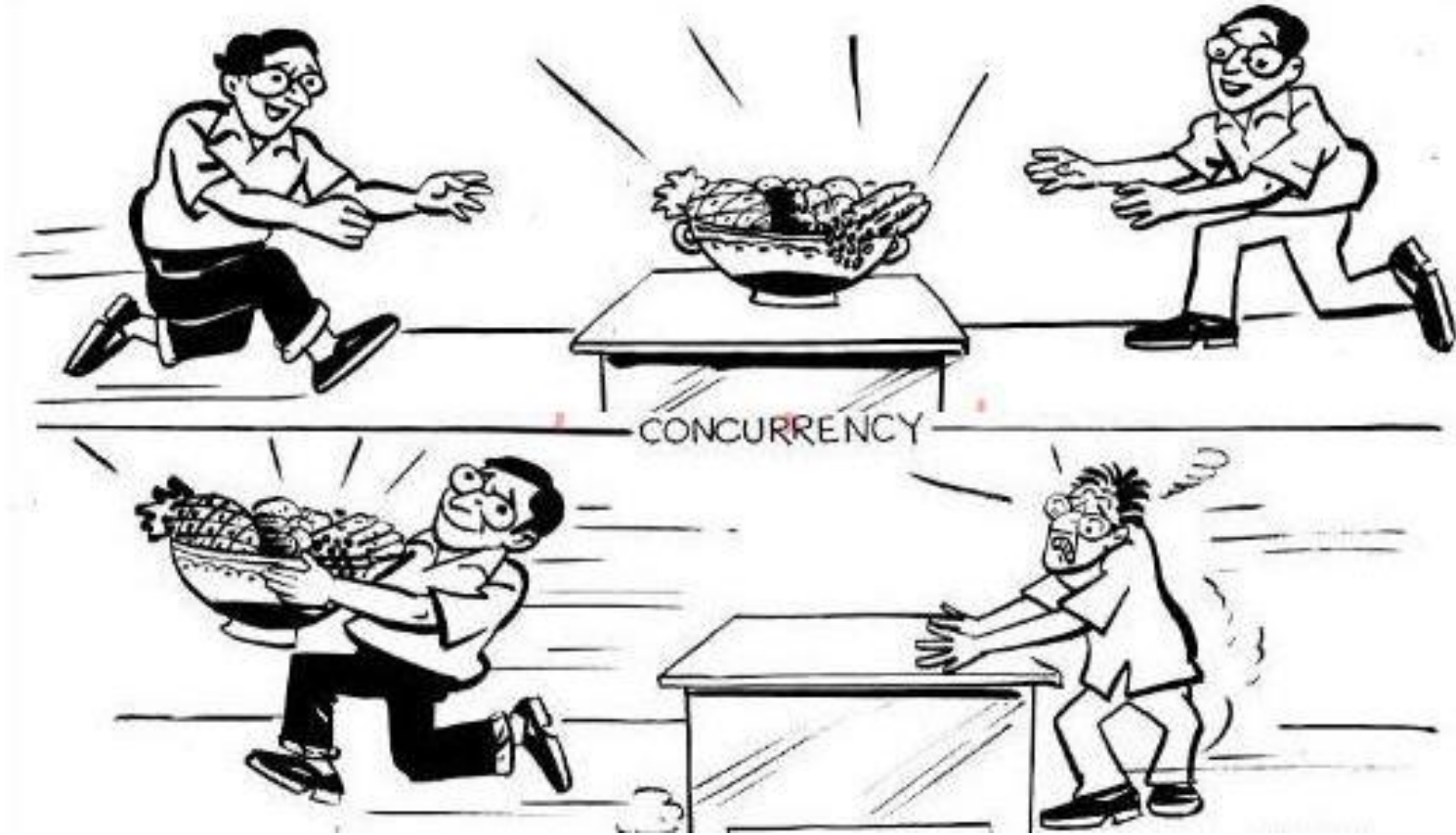
### **RELACIONALES**

# CURSORES

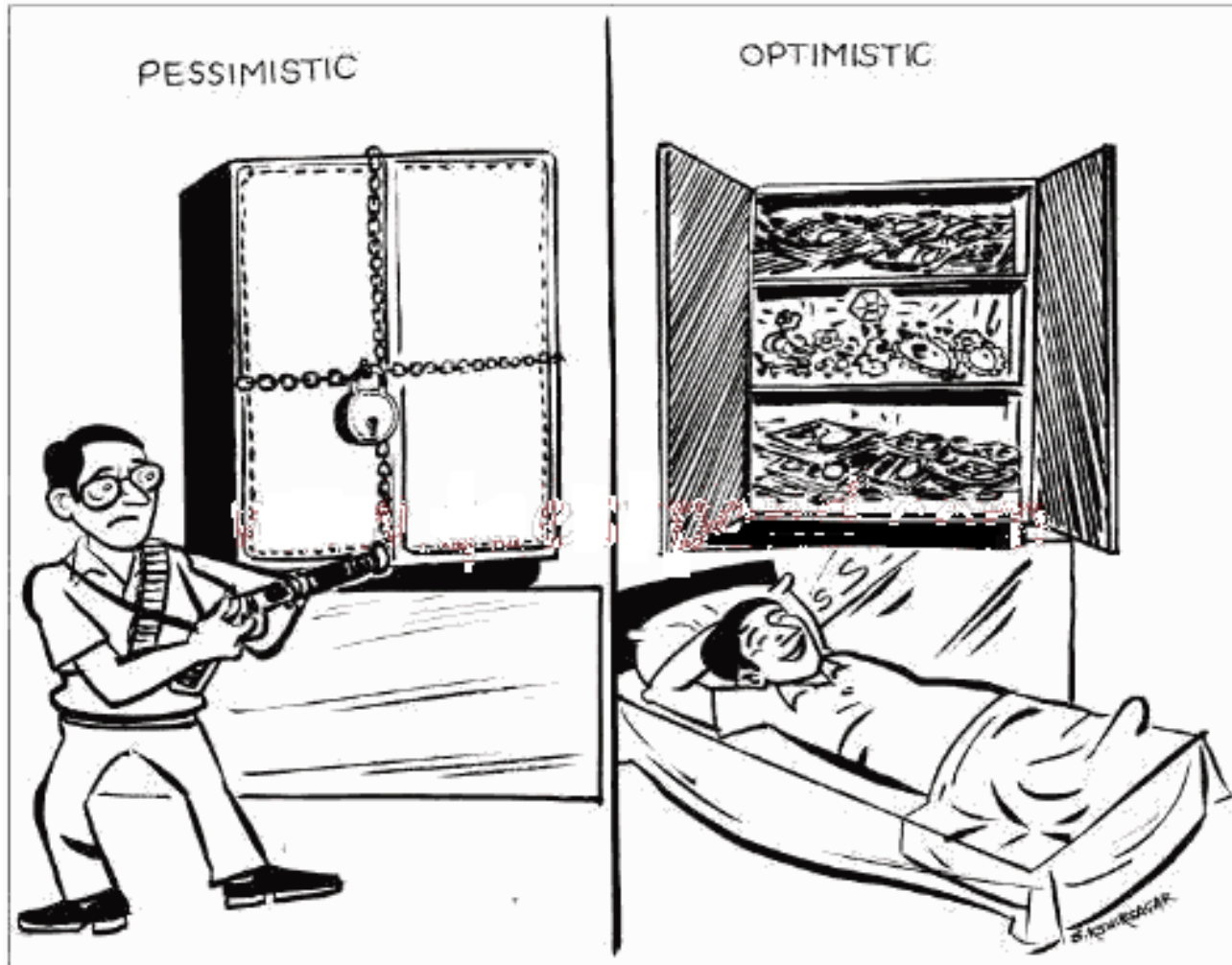




# CONCURRENCIA



# CANDADOS

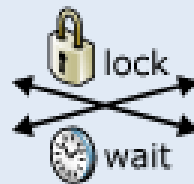


# CANDADOS

## Guaranteed Deadlock

### Transaction 1

Begin transaction  
Update table Supplier  
Update table Part  
Commit transaction



## (Concurrent Transactions)

### Transaction 2

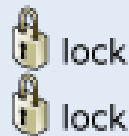
Begin transaction  
Update table Part  
Update table Supplier  
Commit transaction



## No Deadlock

### Transaction 1

Begin transaction  
Update table Supplier  
Update table Part  
Commit transaction



### Transaction 2

Begin transaction  
Update table Supplier  
Update table Part  
Commit transaction



# OPTIMIZACIÓN

Explain Plan x			
SQL   11.67 seconds			
OPERATION	...	CARDINALITY	COST
SELECT STATEMENT		930	953
PX COORDINATOR			
PX SEND (QC (RANDOM))	...	930	953
HASH JOIN (BUFFERED)		930	953
Access Predicates			
NESTED LOOPS		930	953
BUFFER (SORT)			
PX RECEIVE			
PX SEND (HASH)	...		
STATISTICS COLLECTOR			
TABLE ACCESS (BY INDEX ROWID BATCHED)	...	930	23
INDEX (RANGE SCAN)	...	930	8
Access Predicates			
INDEX (UNIQUE SCAN)	...	1	1
Access Predicates			
PX RECEIVE		1	1
PX SEND (HASH)	...	1	1
PX BLOCK (ITERATOR)		1	1
INDEX (FAST FULL SCAN)	...	1	1

**TANTO POR  
APRENDER!**

# **EVALUACIÓN**

# PREGUNTAS SELECCIÓN MÚLTIPLE

Pregunta 1

Sin responder  
aún

Puntúa como  
1,00

Cuáles son las tres principales partes en el Modelo MVC?

Seleccione una o más de una:

- ☐ a. El modelo que representa los datos de la aplicación.
- ☐ b. La vista que presenta los datos.
- ☐ c. El componente que representa las clases de la aplicación.
- ☐ d. El controlador que maneja y en ruta las peticiones de los clientes.
- ☐ e. El módulo que representa una parte de la aplicación.

Penalización

Calificación

El módulo que representa una parte de la aplicación.

-50%

Retroalimentación

↓

i

B

I

Ff

T

💡

✎

💧

Elección 5

↓

i

B

I

Ff

T

💡

✎

💧

El componente que representa las clases de la aplicación

Calificación

-50%

