Bases de Datos

Orientación a objetos

A dónde vamos, de dónde venimos

del modelo relacional a la complejidad

1. Orientación a objetos
2. Datos no estructurados
3. Recuperación de Información
4. Data Mining
5. Data Warehousing

Lecturas

Apartado 2.7 Silberschatz

Capítulo 8 y 9 Silberschatz

https://es.wikipedia.org/wiki/L%C3%B3gica_de_descripci%C3%B3n

Fortalezas del modelo relacional

1. ?
2. ?
3. ?

Limitaciones del modelo relacional

- Representaciones forzadas para algunos problemas
- Escenarios con relaciones complejas
- Escenarios cambiantes
- Inferencias

Solución

División entre interfaz e implementación

- el esquema de la BD sólo tiene interfaz
- **Variables** (atributos en relacional)
- **Mensajes** que se responden con métodos

Entidades como objetos: ejemplo

atributo *dirección* de *empleado*

- variable dirección
- mensaje obtener-dirección
- mensaje establecer-dirección, con parámetro nueva-dirección

Solución

División entre interfaz e implementación

- el esquema de la BD sólo tiene interfaz
- **Variables** (atributos en relacional)
- **Mensajes** que se responden con métodos

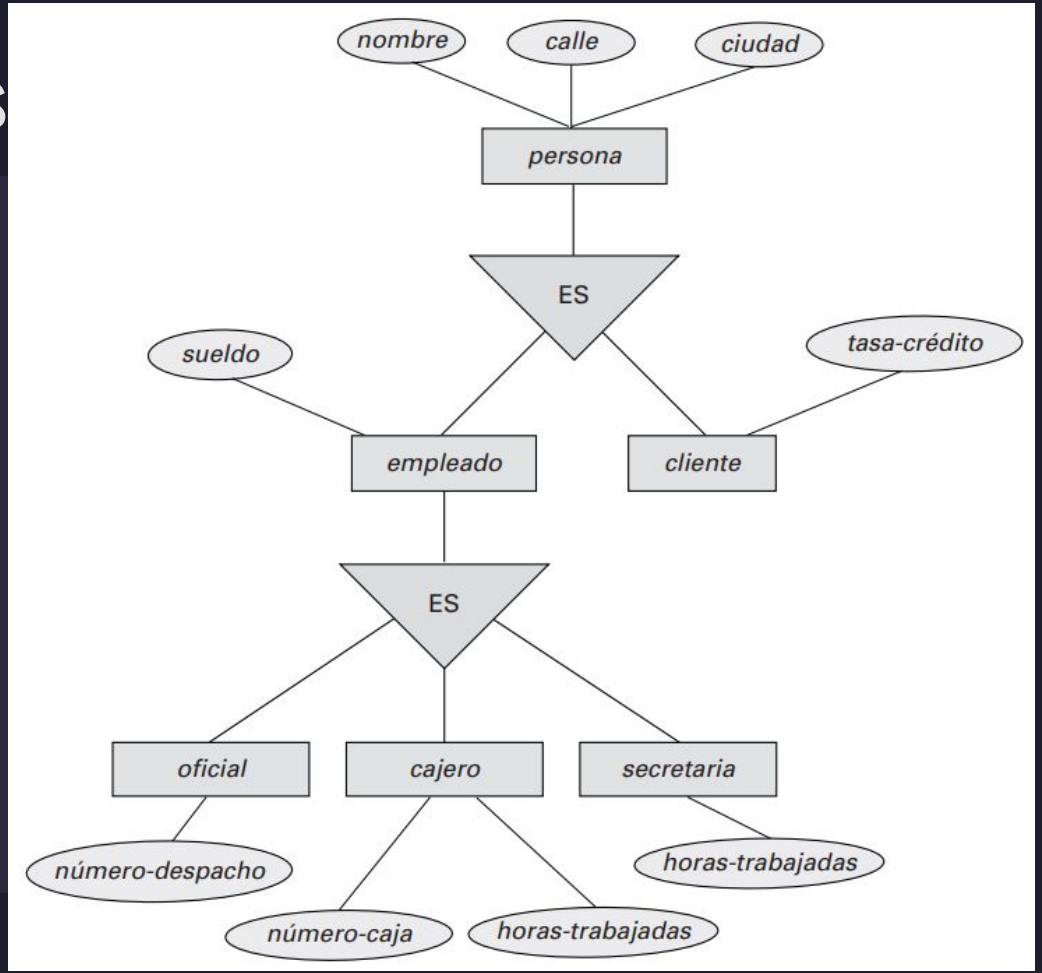
y **Herencia**!

Generalización y especialización

Especialización: un subconjunto de entidades con atributos particulares (diseño descendente)

Generalización: un conjunto de entidades con atributos comunes a varios conjuntos (diseño ascendente)

Generalización y es



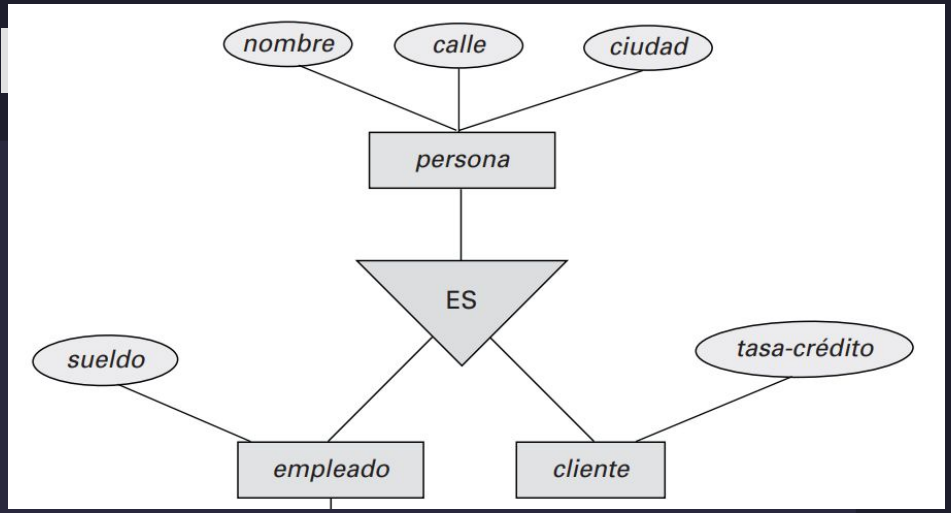
Generalización y especialización

- **Herencia** de atributos (también múltiple)
- Jerarquía de entidades
- **Disjuntas** o **solapadas**
- **Total** (cada entidad del nivel alto debe pertenecer a un conjunto de entidades del nivel bajo) vs. **parcial**

Reducción a tablas de la especialización

1. una tabla para la entidad de nivel alto
2. Para las entidades de nivel más bajo, una tabla con una columna para cada atributo y una columna para cada atributo de la clave primaria de la entidad de nivel más alto

Reducción a tablas de



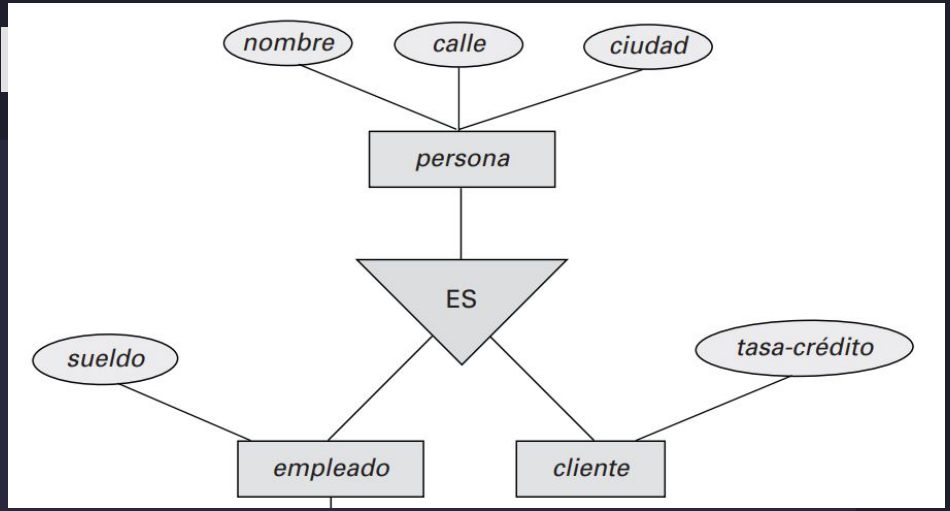
- persona, con nombre, calle y ciudad
- empleado, con nombre y sueldo
- cliente, con nombre, tasa-crédito

Reducción a tablas de la especialización

Para especialización disjunta y completa

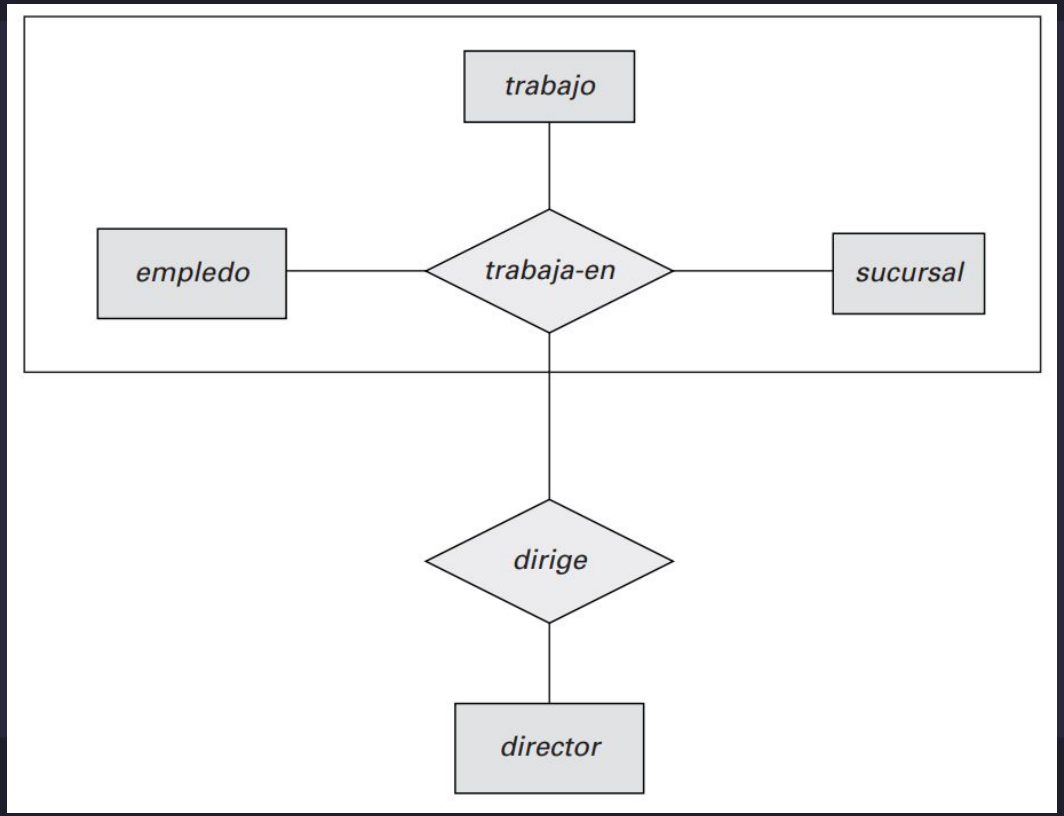
1. ~~una tabla para la entidad de nivel alto~~
2. Para las entidades de nivel más bajo, una tabla con una columna para cada atributo y una columna para **cada atributo** de la entidad de nivel más alto

Reducción a tablas de

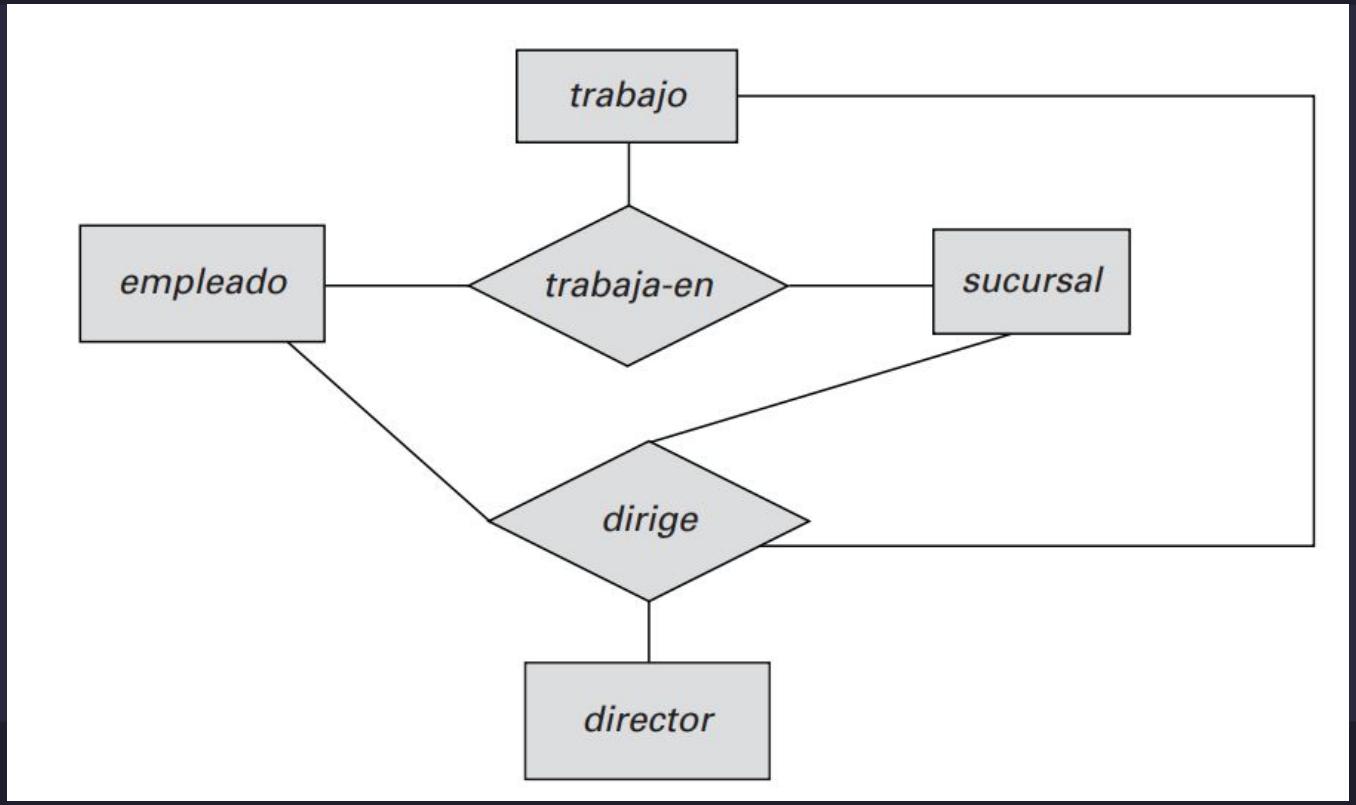


- empleado, con nombre, calle, ciudad y sueldo
- cliente, con nombre, calle, ciudad y tasa-crédito

Reducción a tablas de la agregación

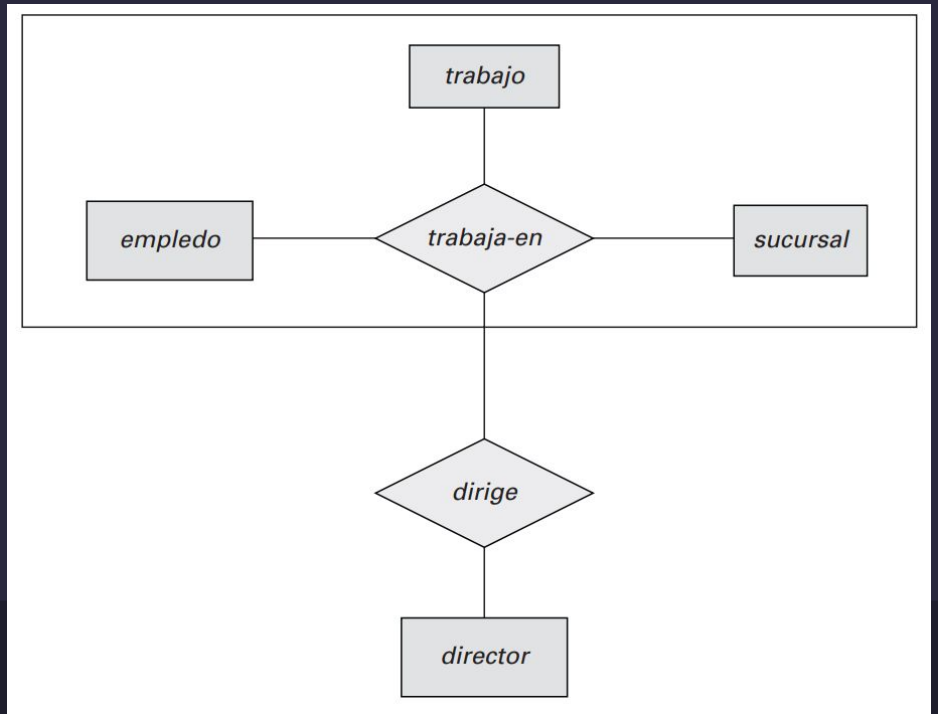


Relaciones entre relaciones

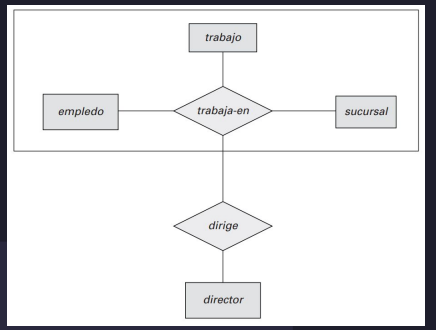


Relaciones entre relaciones: agregación

Representar la relación como entidad



Reducción a tablas de la agregación



La tabla para la relación *dirige*, entre *trabaja-en* y *director*, tiene una columna para cada atributo de la clave primaria de director y de trabaja-en

Recordemos

Lenguaje de descripción

Lenguaje de consulta (procedimental o declarativo)

Esquema

Almacenamiento (Tablas)

Programación orientada a objetos vs. BD

- Interfaz: variables y mensajes
- Clases y subclases, herencia

Programación orientada a objetos vs. BD

- Interfaz: variables y mensajes
- Clases y subclases, herencia
- Clases vs. objetos y **persistencia**

Persistencia

En orientación a objetos, los objetos desaparecen cuando se termina la ejecución

Queremos que los objetos permanezcan en la base de datos!

Persistencia: el estado de un sistema dura más que el proceso que lo creó

Integrar orientación a objetos a las BDs

Dos alternativas:

1. Añadir persistencia a lenguajes de programación orientados a objetos.
2. Extender el modelo de datos relacional con un sistema de tipos más rico

Lenguajes orientados a objetos con persistencia

Java, C++, Smalltalk, con persistencia

- Por clases
- Por método de creación
- Por marcas
- Por alcance

https://en.wikipedia.org/wiki/Comparison_of_object_database_management_systems

¿Cómo se recuperan los objetos?

- Por nombre (como archivos): no escala
- Con punteros persistentes, incluso físicos
- En colecciones iterables

Lenguajes de consulta **procedimentales**!

Lenguajes relacionales con tipos complejos

- SQL con tipos complejos y programación orientada a objetos (capítulo 9)

conservar los fundamentos relacionales (consultas declarativas) mejorando las capacidades de modelado

Modelo relacional anidado

Los atributos pueden tener una relación (no solamente valores atómicos)

¡Los objetos complejos pueden representarse con una única tupla!

Tipos complejos

Tipos complejos y estructurados

```
create type Editorial as
        (nombre varchar(20),
        sucursal varchar(20))
create type Libro as
        (título varchar(20),
        array-autores varchar(20) array [10],
        fecha-pub date,
        editorial Editorial,
        lista-palabras-clave setof(varchar(20)))
create table libros of type Libro
```

Herencia

```
create type Estudiante
        under Persona
        (curso varchar(20),
        departamento varchar(20))
create type Profesor
        under Persona
        (sueldo integer,
        departamento varchar(20))
```

Herencia y herencia de tablas

```
create type Estudiante
        under Persona
        (curso var
        departame
create type Pro
        under Per
        (sueldo integer,
        departamento varchar(20))
```

```
create table estudiantes of Estudiante
        under persona
create table profesores of Profesor
        under persona
```

Herencia y herencia múltiple de tablas

```
create type Estudiante
        under Persona
        (curso var
        departame
create type Pro
        under Per
        (sueldo integer,
        departamento
```

```
create table estudiantes of Estudiante
        under persona
create table profesores of Profesor
        under persona
```

```
create table ayudantes
of Ayudante
        under estudiantes, profesores
```

BDs relacionales OO (no BDs OO)

Como una BD relacional pero con un modelo orientado a objetos: los esquemas y el lenguaje de consulta proveen objetos, clases y herencia directamente, así como extensión del modelo de datos con tipos y métodos personalizados.

https://en.wikipedia.org/wiki/Object%E2%80%93relational_database

Paseo por algunos lenguajes

XML

RDF

json

bson

XQuery

SPARQL

Paseo por algunos lenguajes

XML

RDF

json

bson

XQuery

SPARQL

Paseo por algunos lenguajes

XML	estándar para el intercambio de información estructurada entre diferentes plataformas
RDF	
json	tecnología sencilla con tecnologías asociadas, ecosistema muy grande
bson	
XQuery	
SPARQL	

Paseo por algunos lenguajes

XML

RDF

json

bson

XQuery

SPARQL

Modelo de datos para descripción conceptual en recursos web

Paseo por algunos lenguajes

XML

RDF

json

bson

XQuery

SPARQL

formato de texto sencillo
para el intercambio de datos

Paseo por algunos lenguajes

XML

RDF

json

bson

XQuery

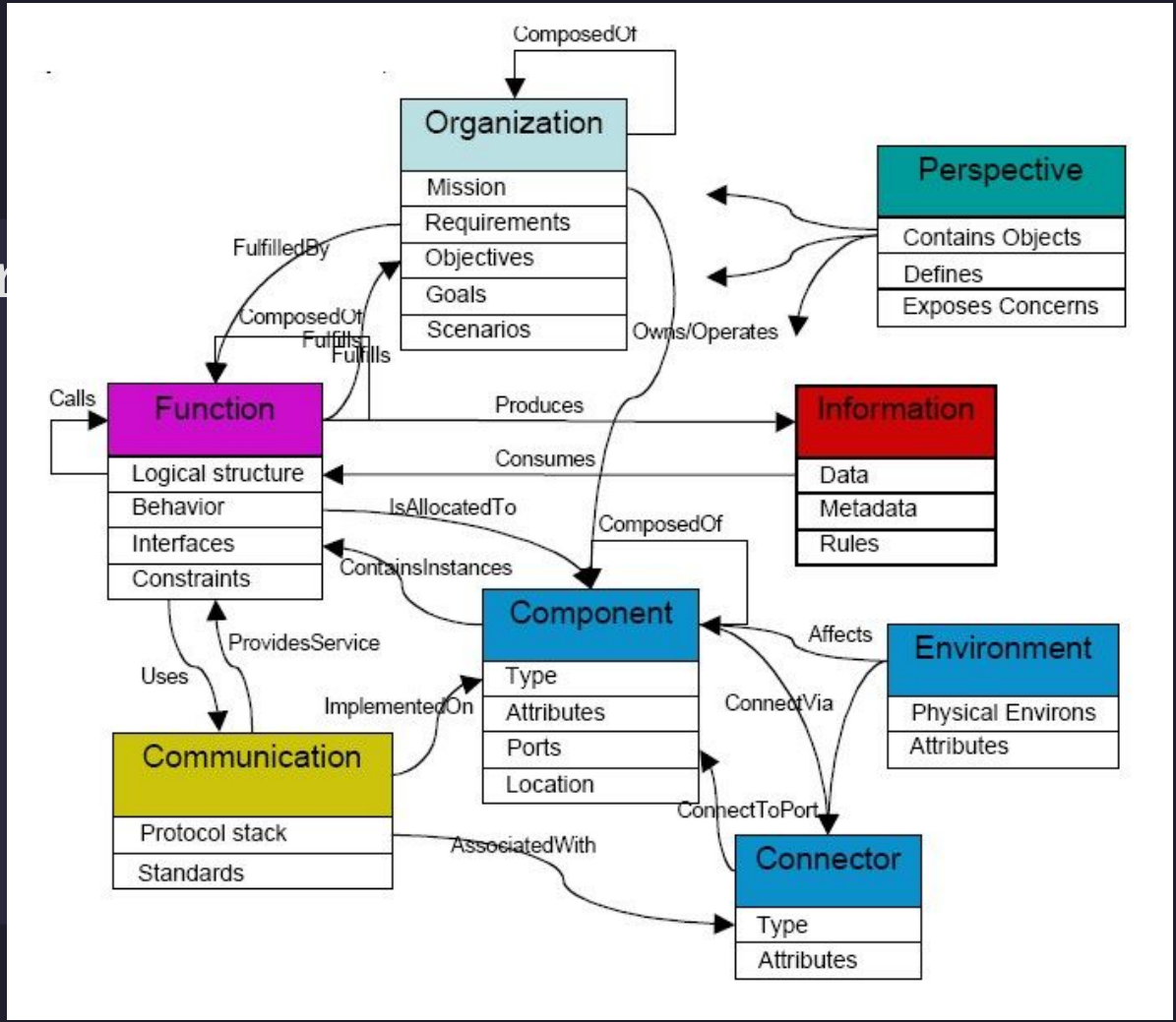
SPARQL OWL

Ontologías

Grafos de representación del conocimiento

Ontologías

Grafos de repr



Lógicas de descripción

lenguajes de representación del conocimiento

- descripciones de conceptos (de dominio)
- semántica: equivalencia entre fórmulas de lógicas de descripción y expresiones en lógica de predicados de primer orden

https://es.wikipedia.org/wiki/L%C3%B3gica_de_descripci%C3%B3n

Lógicas de descripción

- formalismo descriptivo: conceptos, roles, individuos y constructores.
- formalismo terminológico: descripciones complejas y propiedades
- formalismo asertivo: propiedades de individuos

https://es.wikipedia.org/wiki/L%C3%B3gica_de_descripci%C3%B3n

Qué se puede hacer con lógicas de descripción

- Cuantificación, restricción de dominio
- Negación
- Reciprocidad
- Transitividad

Qué se puede hacer con lógicas de descripción

- Cuantificación, restricción de dominio
- Negación
- Reciproci
- Transitiv

Axiomas	Semántica	Ejemplo
$\geq n\ R$	$\{x \mid \#\{y \mid R^{\mathcal{I}}(x,y)\} \geq n\}$	$\geq 2\ tieneHijo$
$\leq n\ R$	$\{x \mid \#\{y \mid R^{\mathcal{I}}(x,y)\} \leq n\}$	$\leq 2\ tieneHijo$
$\geq n\ R.C$	$\{x \mid \#\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} \geq n\}$	$\geq 3\ tieneHijo.Hombre$
$\leq n\ R.C$	$\{x \mid \#\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} \leq n\}$	$\leq 3\ tieneHijo.Mujer$
$a_1 \ldots a_n$	$\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$	$\{Maria, John\}$
$R \sqcup S$	$R^{\mathcal{I}} \cup S^{\mathcal{I}}$	$tieneHijo \sqcup tieneHija$
$R \sqcap S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}}$	$tieneHijo \sqcap tieneHija$
$\neg R$	$(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \mid R^{\mathcal{I}}$	$\neg tieneAmigo$

Lógicas de descripción

- inferir nuevo conocimiento con algoritmos de razonamiento decidibles… más o menos: http://www.cs.man.ac.uk/~ezolin/dl/

https://es.wikipedia.org/wiki/L%C3%B3gica_de_descripci%C3%B3n

Desventajas de usar una ontología

Soluciones intermedias

Mapeo entre ontología y BD relacional

Y en todo esto, dónde entra MongoDB?

Masticando conceptos

Datos no estructurados

Texto

Imágenes

Secuencias temporales

Perfiles de personas

Bases de datos no relacionales (not only SQL)

- orientadas a documentos
- key-value
- wide column
- Grafos

filminas del laboratorio sobre NoSQL

/THANKS!

/DO YOU HAVE ANY QUESTIONS?

youremail@freepik.com
+91 620 421 838
yourwebsite.com