

Mongo -- Review(ish)

Mongo 1-1

- Very effective at querying within a collection
 - Not as good at Joins (because a properly setup mongo DB rarely needs joins)
- So set up collections to avoid joins
 - Rather than one-to-one relations in separate documents embedded one document within the other

Cars			Engines			
_id	owner	make	_id	car_id	power	consumption
007	Daniel	Ferrari	234808	007	660	10
008	Daniel	Fiat	008	008	120	45

OR

Cars				
_id	owner	make	power	consumption
007	Daniel	Ferrari	660	10
008	Daniel	Fiat	120	45

```
{
  "_id": 007,
  "owner": "Daniel",
  "make": "Ferrari",
  "engine": {
    "power": 660hp,
    "consumption": 10mpg
  }
}
```

Mongo 1-Many

- For 1-N put embedded documents into an array

Cars		
_id	owner	make
007	Daniel	Ferrari
008	Daniel	Fiat

Wheels	
_id	car_id
234819	007
281928	007
392838	007
928038	007
950555	008

```
{
  "_id": 007,
  "owner": "Daniel",
  "make": "Ferrari",
  "wheels": [
    { "partNo": 234819 },
    { "partNo": 281928 },
    { "partNo": 392838 },
    { "partNo": 928038 }
  ],
  ...
}
```

- Works well when N is small and relatively static

Many-Many

- Examine use cases, if data is always access from one side, then do exactly as 1-Many.
- If need to query from both sides, accept data duplication and represent both sides of the Many-Many is both places.

Actor Collection

```
{"actor":actorID, "name":"Mary",  
  "films":[f1, f2, f3],  
  ...}
```

Film Collection

```
{"film":filmID,  
  "title":"Gone with the Ducks"  
  "actors":[aid1, aid2, aid3],  
  ... }]
```

- Problem. As shown would need to do join to show any info on a film given an actor.
 - If represent all of actor by embedding in film and ll of film embedded within actor, then have a recursive dependency.

Subset and extended references

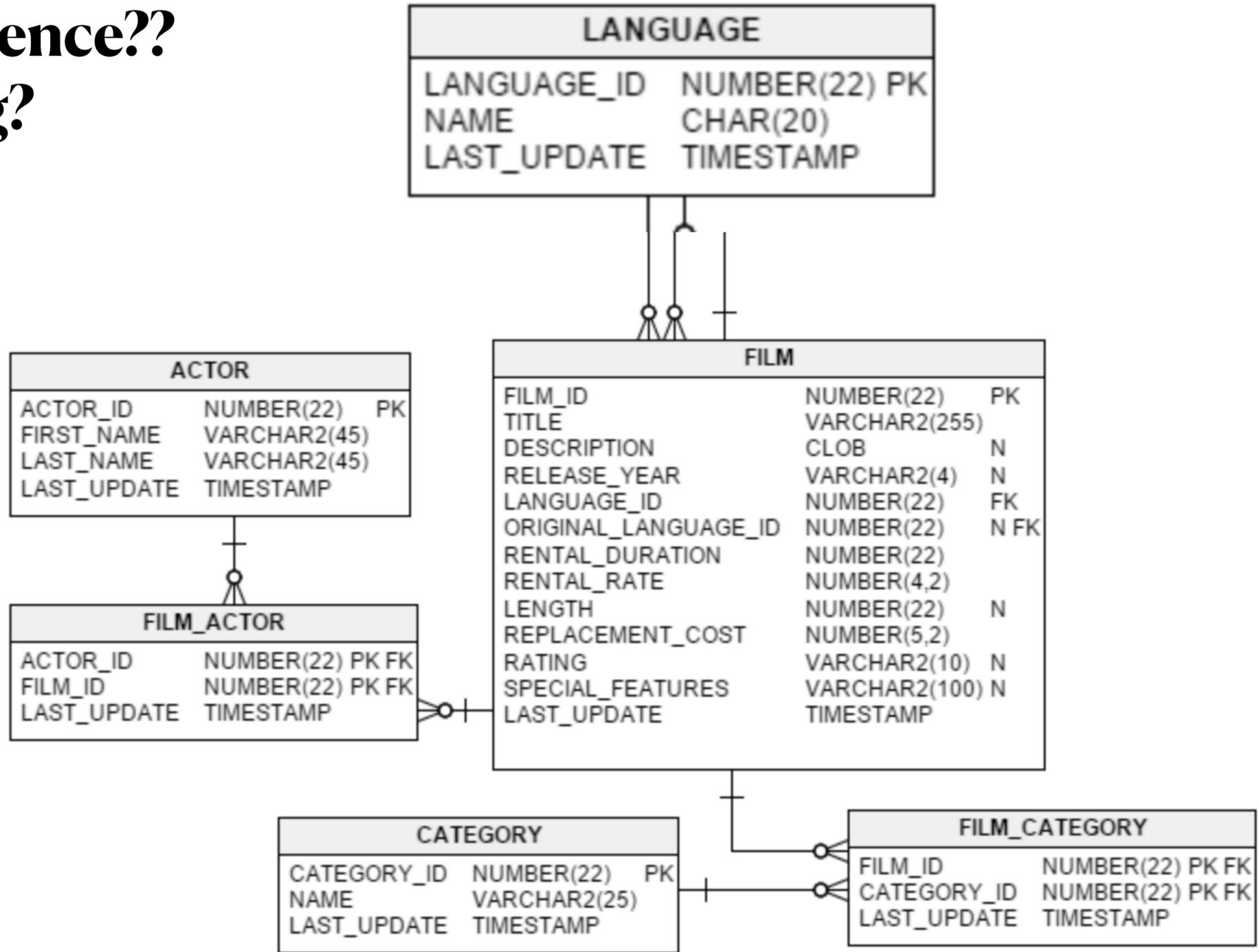
- Embedded subset
 - A one to many relation but many is large and possibly changing frequently
 - Idea, embed only some of the many in the one
 - need to occasionally the "some"
- Extended Reference
 - Idea rather than just embed reference to another table, embed reference plus some key data
 - Show that key data and allow drill down from there

Sakila

Embedded Subset??

Extended Reference??

Full Embedding?



My Choice for Sakila in Mongo

```
db.mgoactor.find().limit(2)
[
  {
    _id: ObjectId("624083255617ccf3e0453c3f"),
    actor: 1,
    first_name: 'PENELOPE',
    last_name: 'GUINNESS',
    films: [
      {
        categ: 2, catname: 'Animation',
        filmid: 23, filmname: 'ANACONDA CONFESSIONS'
      },
      {
        categ: 3, catname: 'Children',
        filmid: 509, filmname: 'LANGUAGE COWBOY'
      }, ...
    ]
  }
]
```

```
sakila==> db.mgofilm.findOne()
{
  _id: ObjectId("6240ec934ff1d4bc11fa36ca"),
  film_id: 1,
  title: 'ACADEMY DINOSAUR',
  cat_id: 6,
  cat_name: 'Documentary',
  description: 'A Epic Drama of a Feminist And a Mad Scientist who',
  release_year: 2006,
  language_id: 1,
  language: 'English',
  rental_rate: 0.99,
  length: 86,
  actors: [
    { aid: 1, first_name: 'PENELOPE', last_name: 'GUINNESS' },
    { aid: 10, first_name: 'CHRISTIAN', last_name: 'GABLE' },
    { aid: 20, first_name: 'LUCILLE', last_name: 'TRACY' },
    { aid: 30, first_name: 'SANDRA', last_name: 'PECK' },
    { aid: 40, first_name: 'JOHNNY', last_name: 'CAGE' },
    { aid: 53, first_name: 'MENA', last_name: 'TEMPLE' },
    { aid: 108, first_name: 'WARREN', last_name: 'NOLTE' },
    { aid: 162, first_name: 'OPRAH', last_name: 'KILMER' },
    { aid: 188, first_name: 'ROCK', last_name: 'DUKAKIS' },
    { aid: 198, first_name: 'MARY', last_name: 'KEITEL' }
  ]
}
```

Mongo Query

- Always start "db."
- Next is always the name of the collection
- find the the first and last name of all actors who have been in a category 2 or category 3 movie with someone whose name starts PEN
 - `db.mgofilm.find({ "actors.first_name": { $regex: "^PEN" } })`
 - `db.mgofilm.find({ "actors.first_name": { $regex: "^PEN" }, $or: [{ cat_id: 2 }, { cat_id: 3 }] }, { "actors.$": 1, title: 1, cat_id:1 })`
 - `db.mgoactor.find({first_name:{$regex:"PEN"}, $or:[{"films.categ":2}, {"films.categ":3}]}, {"films.filmname.$":1, first_name:1, _id:0})`