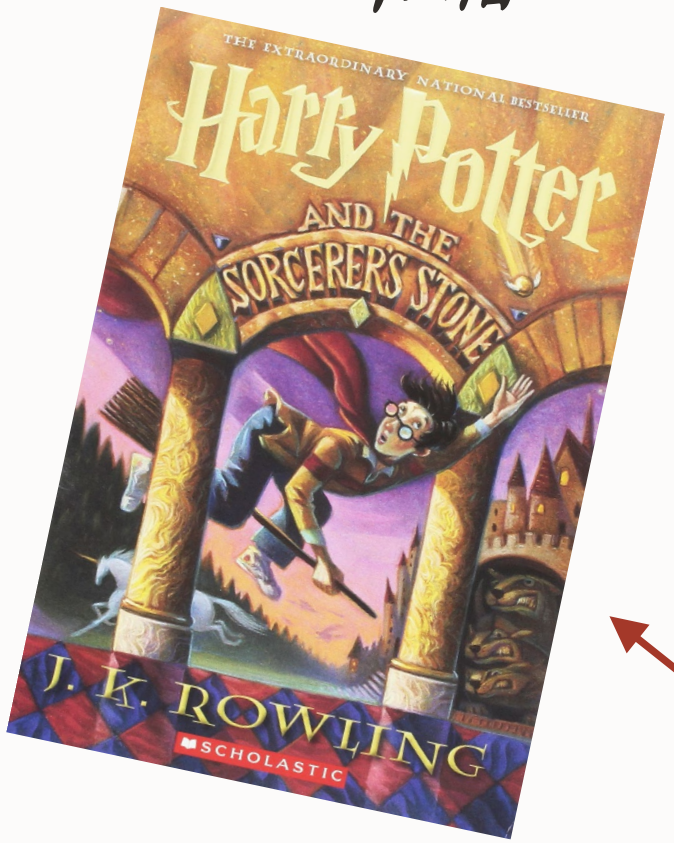
JSON  热，笑傲江湖！

这是真的吗？

张弘弢
资深售前顾问
2022年1月14日

例子：网上商城

书籍

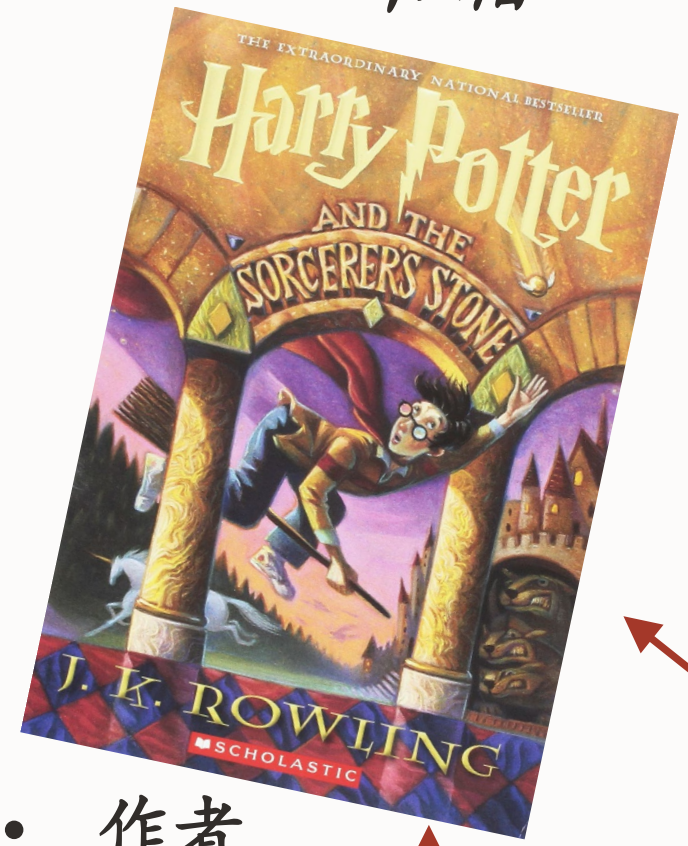


电影

- 名称
- 类型
- 价格
- 数量
- 状况
- 详细描述

例子：网上商城

书籍



- 作者
- 语言

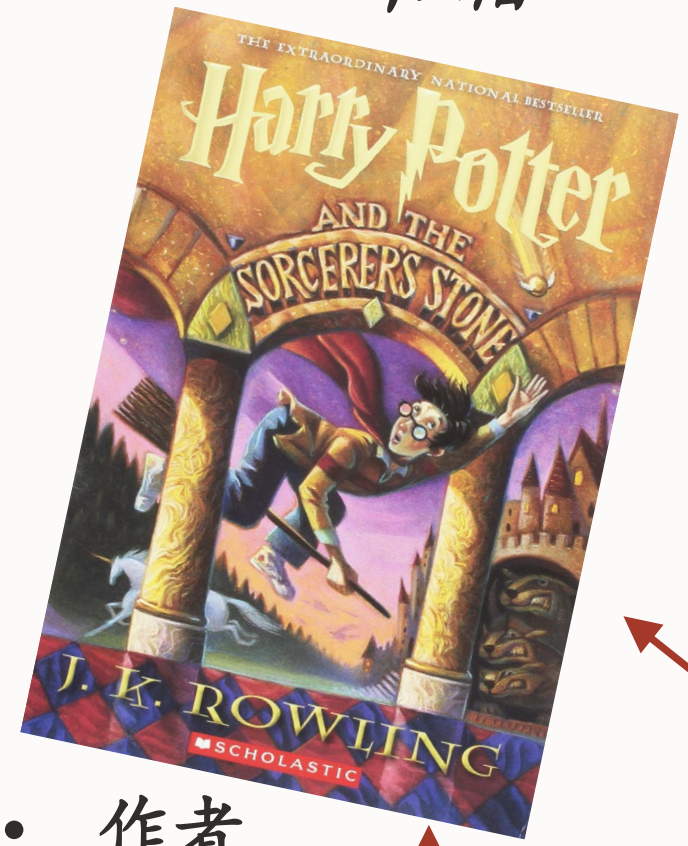
- 名称
- 类型
- 价格
- 数量
- 状况
- 详细描述

电影

- 格式 (DVD, 蓝光)
- 演员
- 语言

例子：网上商城

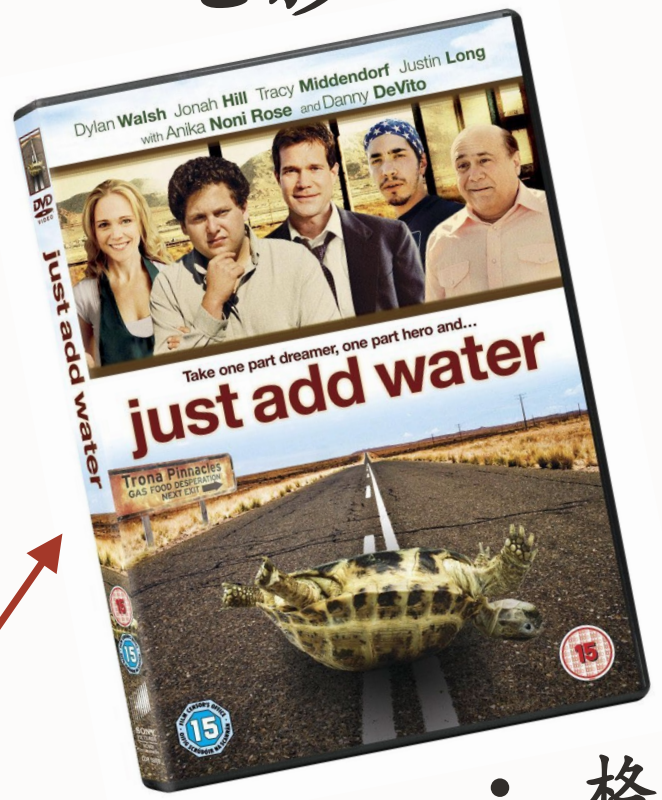
书籍



- 作者
- 语言

- 名称
- 类型
- 价格
- 数量
- 状况
- 详细描述

电影



- 格式 (DVD, 蓝光)
- 演员
- 语言

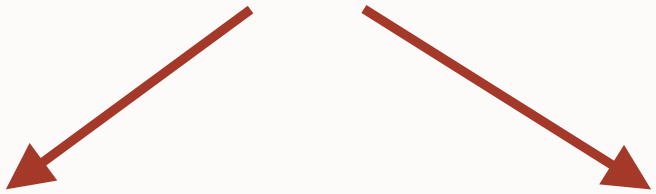
更多

?

- ?

编程语言中的对象

```
public abstract class Product {
    Integer  id;
    String   title;
    Currency price;
    Integer  quantity;
    String   condition;
    String   description;
}
```



```
public class Book extends Product{
    String author;
    String language;
}
```

```
public class Movie extends Product {
    String format;
    List<String> actors;
    List<String> languages;
}
```

范式化和表对象

Products

id	type	Title	Price	Quantity	Condition	Description
	Book					
	Movie					

Books

id	Author	LangID

Movies

id	Format

Actors

aid	Name

Languages

lid	Language

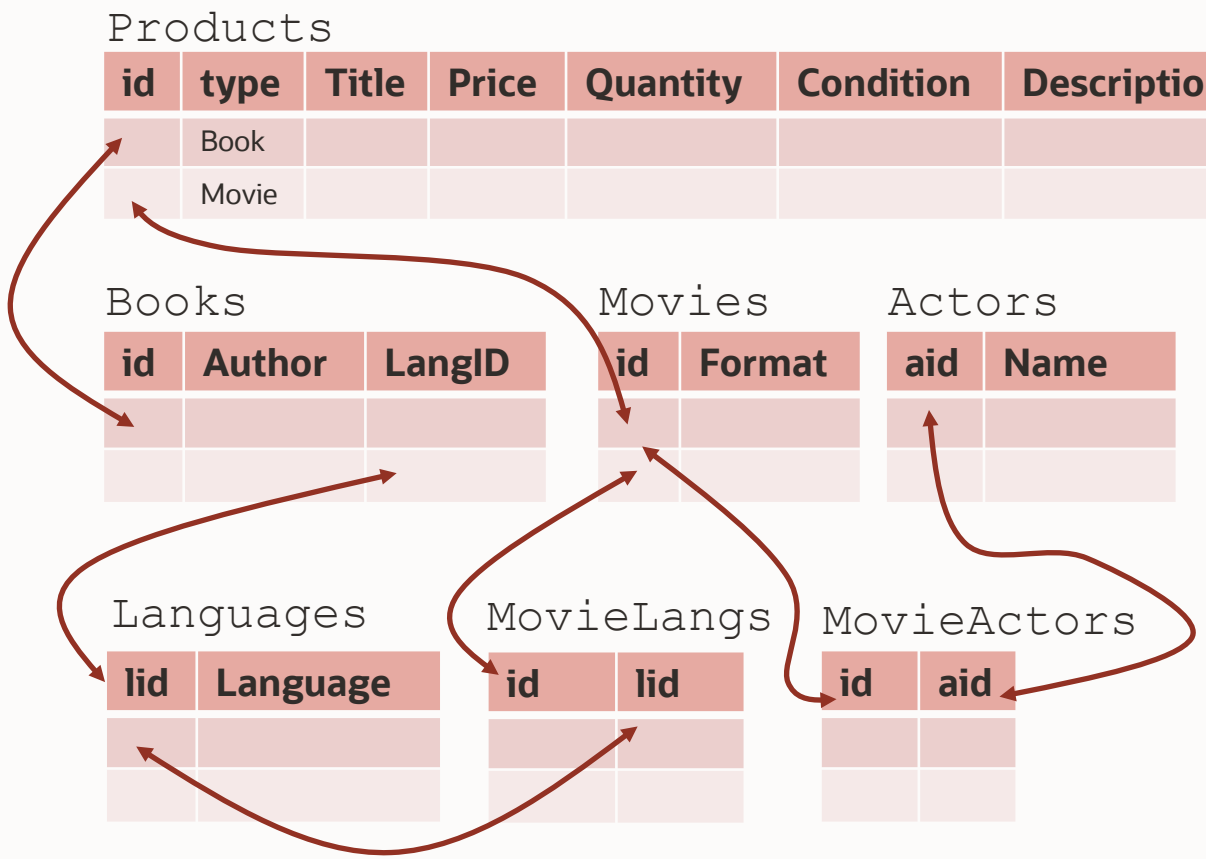
MovieLangs

id	lid

MovieActors

id	aid

范式化和表对象



主键/外键: 数据的完整性/一致性(连接用列)

范式化和表对象

Products

id	type	Title	Price	Quantity	Condition	Description

```
create table "products" (…);
create table "books" (…);
create table …
…
alter table "MovieActors"
  add constraint "movieActorConstr"
  foreign key ("aid")
  references Actors("aid");
…
```

主键/外键: 数据的完整性/一致性(连接用列)

使用JSON集合

Products

id	type	Title	Price	Quantity	Condition	Description

```
create table "products" (…);
create table "books" (…);
create table …
…
alter table "MovieActors"
  add constraint "movieActorConstr"
  foreign key ("aid")
  references Actors("aid");
…
```

主键/外键: 数据的完整性/一致性(连接用列)

```
createCollection("products");
```

数据插入

Products

id	type	Title	Price	Quantity	Condition	Description

```
insert into products values
(123,'movie','Just add water', 3.99, 1, 'like
new','Close to…');

insert into movies values
(1, 'DVD');

insert into movieLangs values (..);
insert into Actors values (..);
insert into MovieActors values (..);
```

```
public void storeMovie (Movie m)
{
    //sql magic here
}
```

```
createCollection("products");
```

数据插入

Products

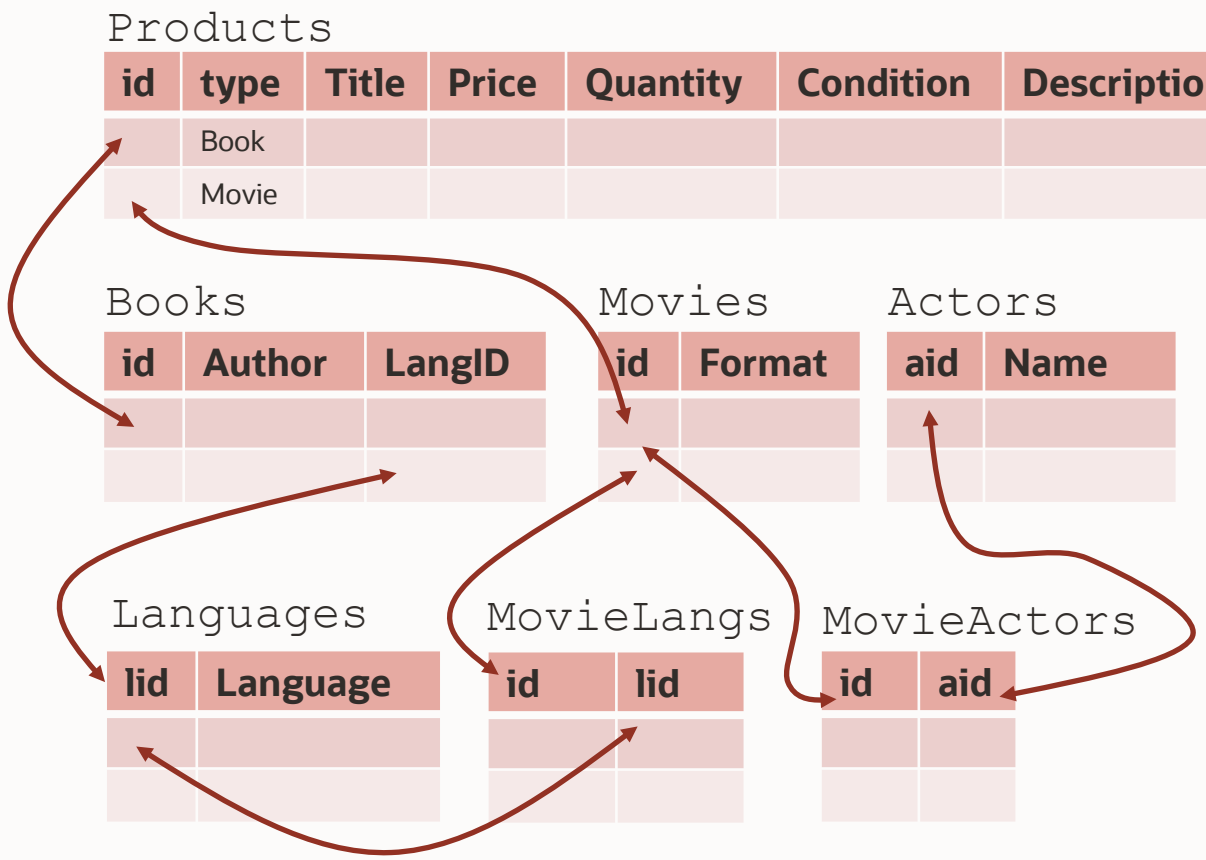
id	type	Title	Price	Quantity	Condition	Description

```
insert into products values
(123,'movie','Just add water', 3.99, 1, 'like
new','Close to…');

insert into movies values
(1, 'DVD');

insert into movieLangs values (..);
insert into Actors values (..);
insert into MovieActors values (..);
```

```
public void storeMovie (Movie m)
{
    //sql magic here
}
```

```
products.insert (
{
  "id": 2827,
  "type": "Movie",
  "title": "Just add water",
  "format": "DVD",
  "quantity": 1,
  "condition": "like new",
  "price": 3.99,
  "languages": ["english","spanish"],
  "actors": ["Danny DeVito","Jonah Hill"],
  "description": "Close to the entrance to
                  Death Valley is the God-
                  forsaken town of Trona…"
}
);
```

链接和内联

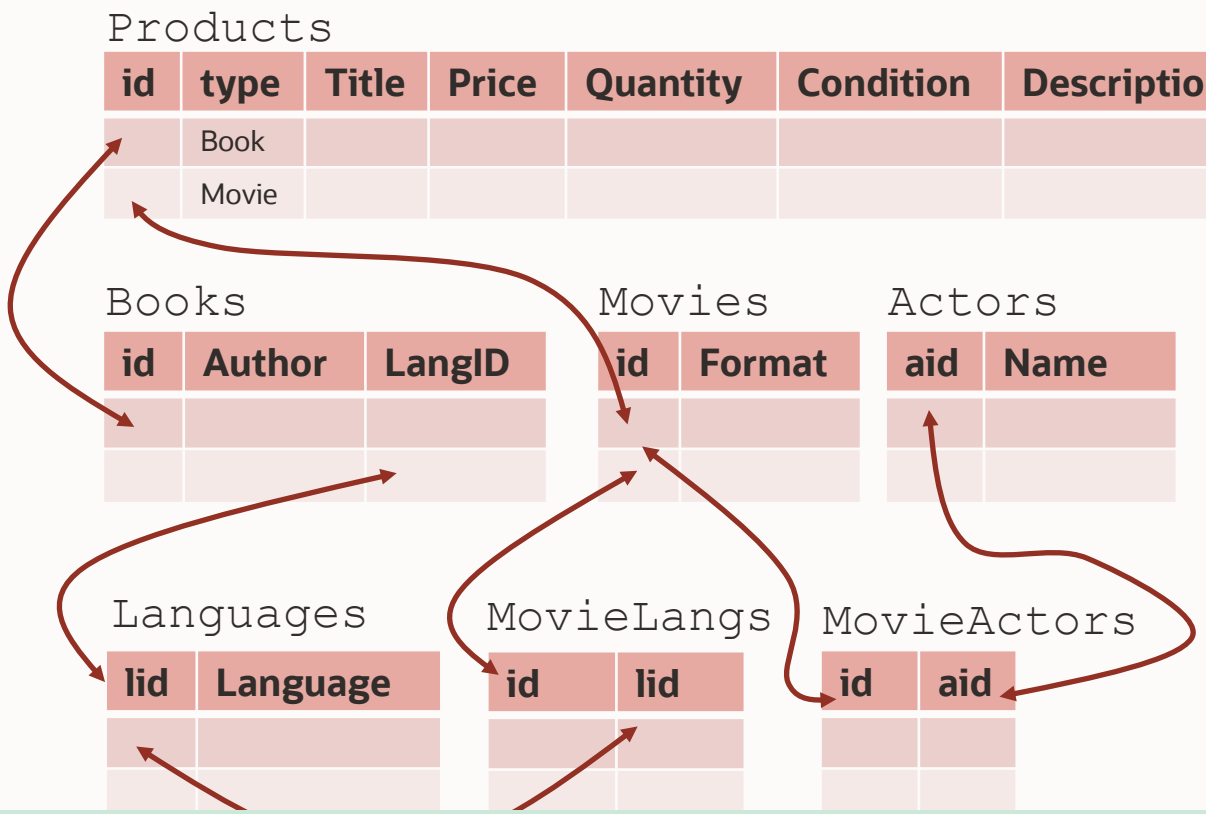



```
products.insert (
{
  "id": 2827,
  "type": "Movie",
  "title": "Just add water",
  "format": "DVD",
  "quantity": 1,
  "condition": "like new",
  "price": 3.99,
  "languages": ["english","spanish"],
  "actors": ["Danny DeVito","Jonah Hill"],
  "description": "Close to the entrance to
                  Death Valley is the God-
                  forsaken town of Trona…"
}
);
```

链接

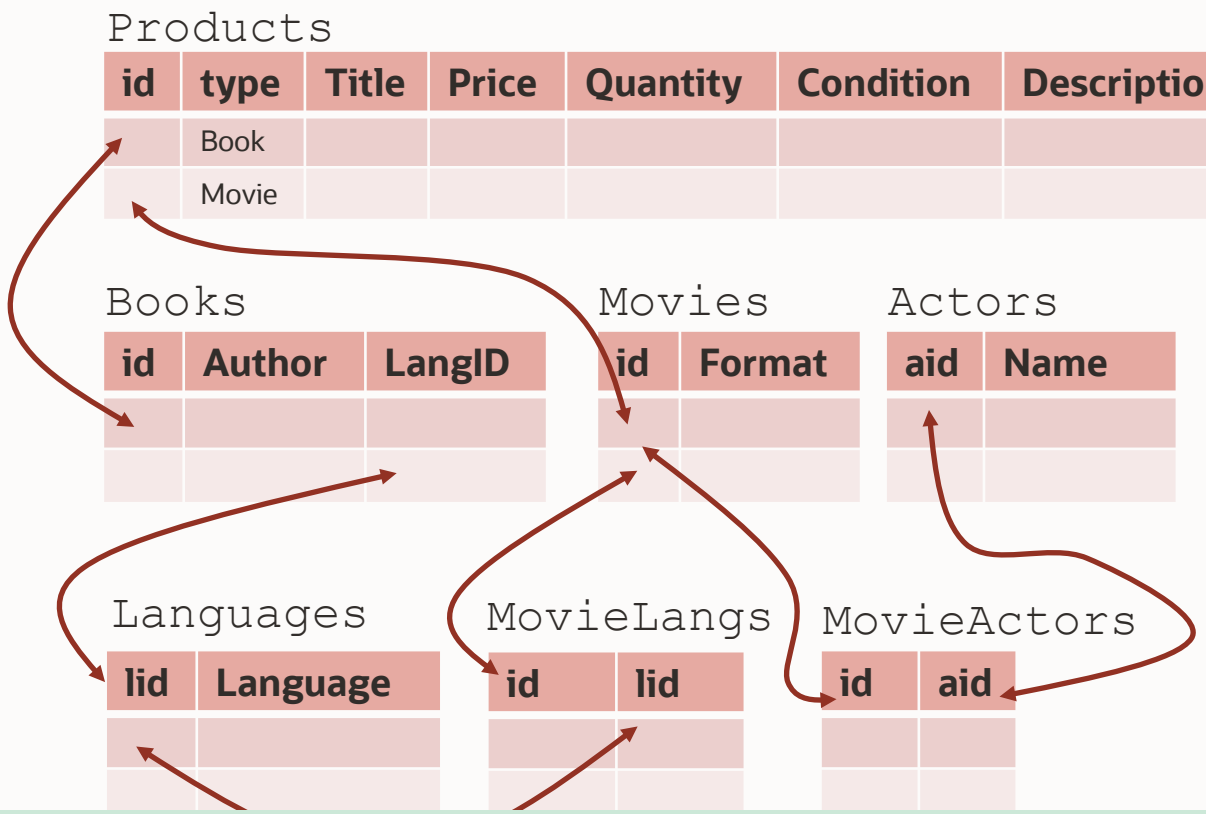
内联

数据检索：按演员查找电影




```
select p.id, p.type, p.title, p.quantity, p.condition, p.description, m.format, a.name
from   Products p, Movies m, Actors a, MovieActors ma
where  p.id = m.id and m.id = ma.id and ma.aid = a.aid and a.name = 'Jonah Hill';
```

数据检索：按演员查找电影

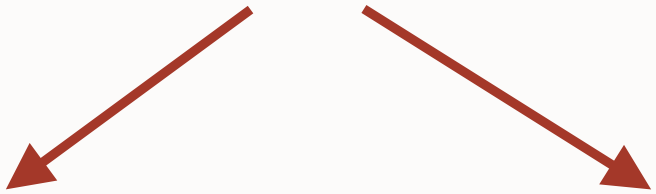



```
products.find('{"actor":"Jonah Hill"}')
```

```
select p.id, p.type, p.title, p.quantity, p.condition, p.description, m.format, a.name
from   Products p, Movies m, Actors a, MovieActors ma
where  p.id = m.id and m.id = ma.id and ma.aid = a.aid and a.name = 'Jonah Hill';
```

应用/模式变更

```
public abstract class Product {
    Integer  id;
    String   title;
    Currency price;
    Integer  quantity;
    String   condition;
    String   description;
}
```

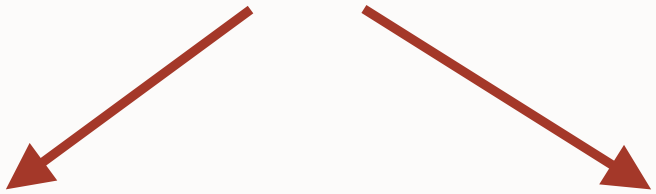


```
public class Book extends Product{
    String author;
    String language;
}
```

```
public class Movie extends Product {
    String format;
    List<String> actors;
    List<String> languages;
}
```

应用/模式变更

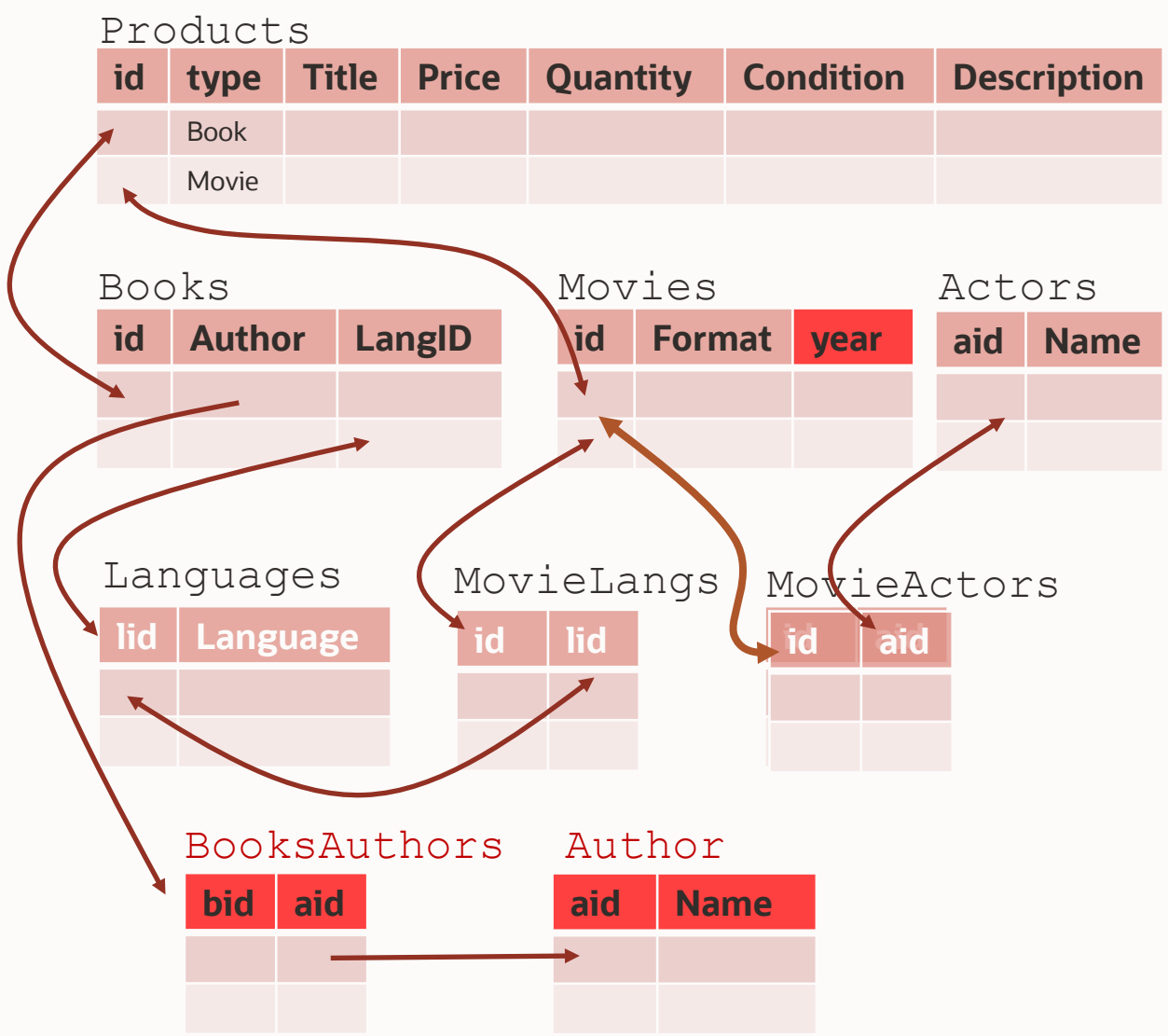
```
public abstract class Product {
    Integer  id;
    String   title;
    Currency price;
    Integer  quantity;
    String   condition;
    String   description;
}
```



```
public class Book extends Product{
    List<String> authors;
    String language;
}
```

```
public class Movie extends Product {
    String format;
    Number yearReleased;
    List<String> actors;
    List<String> languages;
}
```

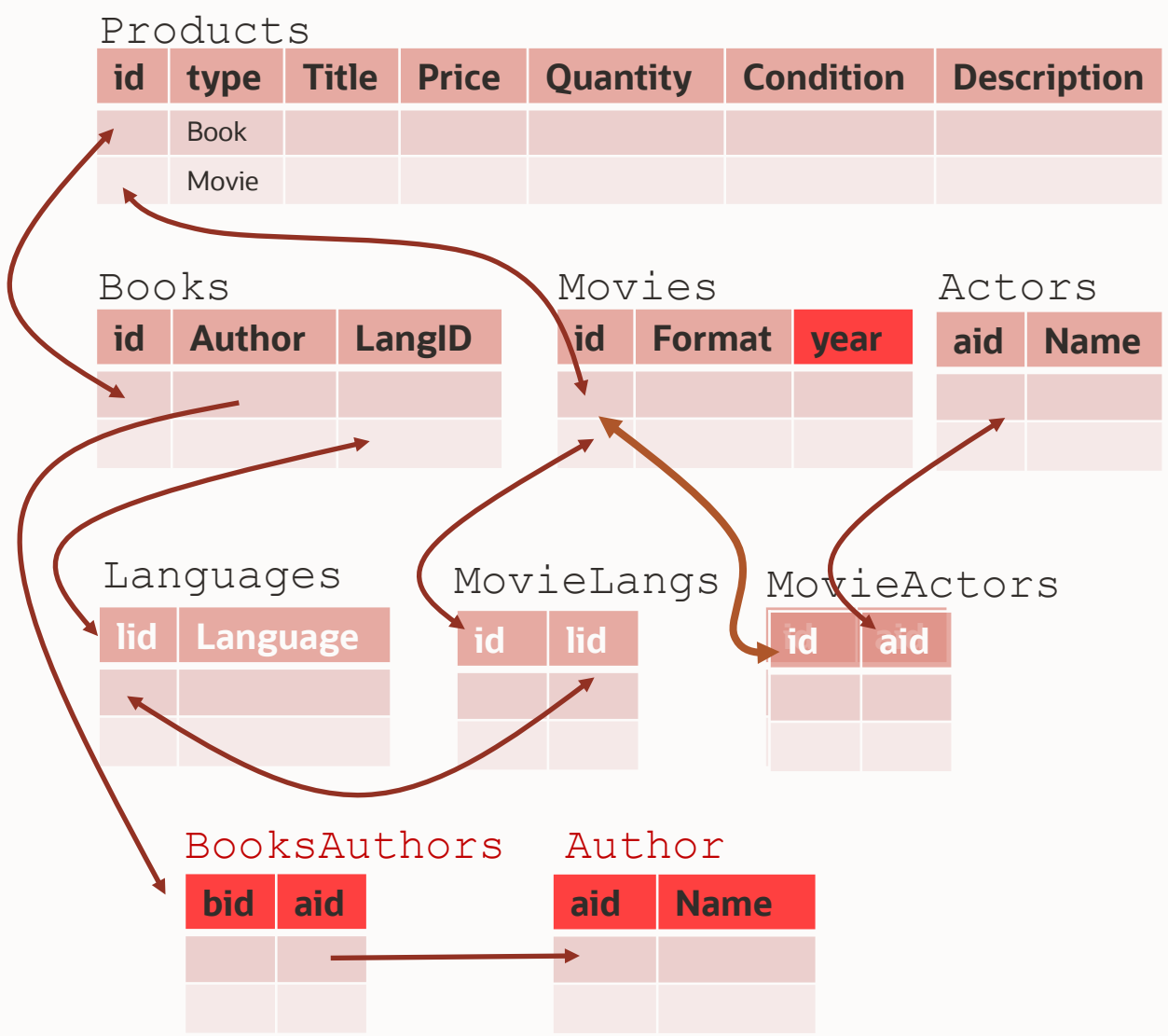
关系模式的修订




```
products.insert (
{
  "type": "Movie",
  "title": "Just add water",
  "format": "DVD",
  "quantity": 1,
  "condition": "like new",
  "price": 3.99,
  "languages": ["english","spanish"],
  "actors": ["Danny DeVito","Jonah Hill"],
  "description": "Close to the entrance to
                  Death Valley is the God-
                  forsaken town of Trona…",
  "released":2008
}
);
```

```
products.find('{"released":2008}')
```

关系模式的修订

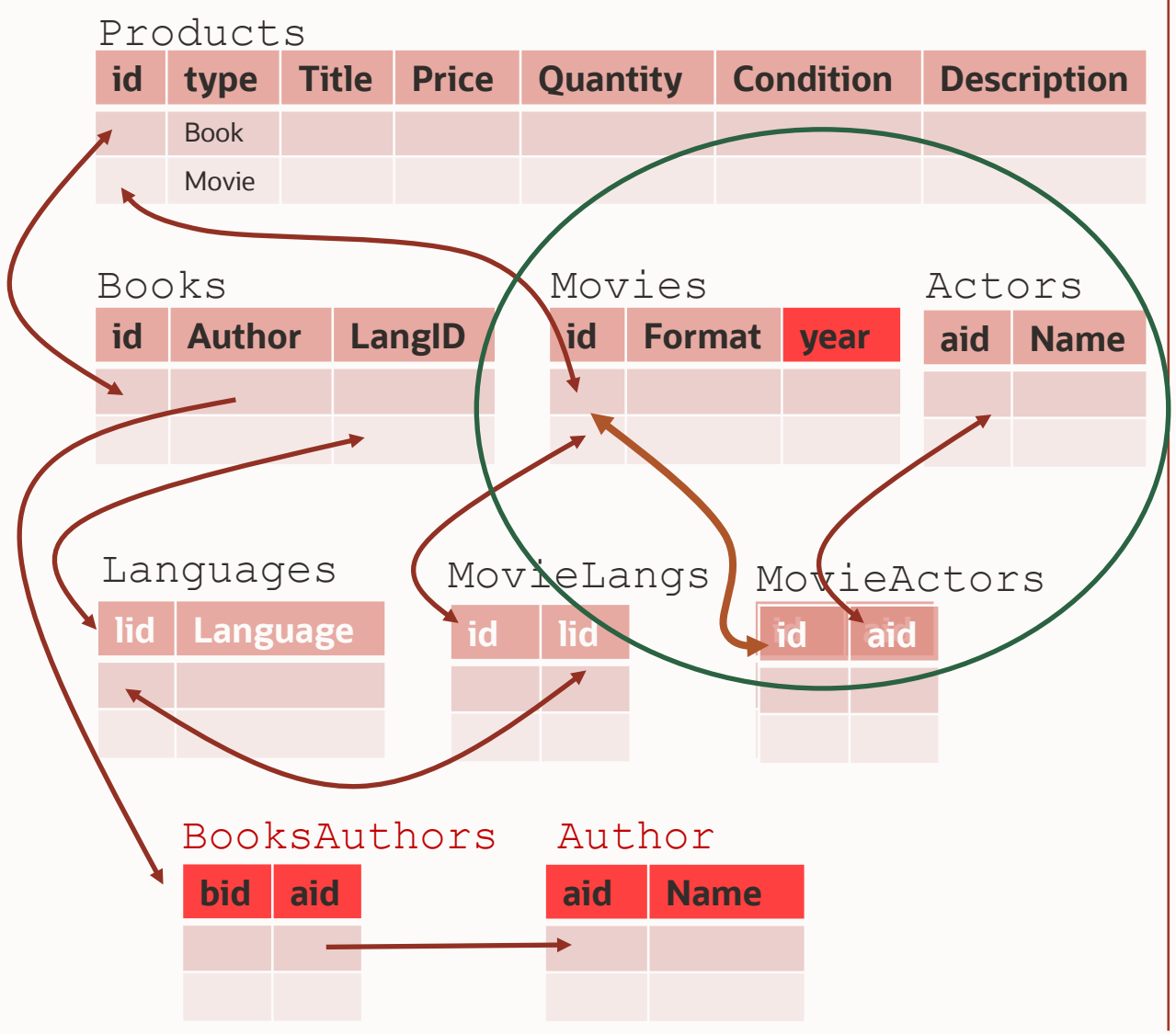



```
products.insert (
{
 "type": "Book",
 "title": "Harry Potter and the Cursed
        Child – The official play script",
 "quantity": 1,
 "condition": "mint",
 "price": 17.05,
 "languages":"english",
 "author": ["J.K Rowling","Jack Thorne",
          "John Tiffany"],
 "description": "hardcover"
}
);
```

```
products.find('{"author":J.K. Rowling}')
```

Ryan McGuire / Gratisography

报表/分析类查询：如 “特定演员的所有电影”




```
products.insert (
{
  "type": "Movie",
  "title": "Just add water",
  "format": "DVD",
  "quantity": 1,
  "condition": "like new",
  "price": 3.99,
  "languages": ["english","spanish"],
  "actors": ["Danny DeVito","Jonah Hill"],
  "description": "Close to the entrance to
                  Death Valley is the God-
                  forsaken town of Trona…",
  "released":2008
}
);
```

内联会导致冗余


```
{
  "type": "Movie",
  "title": "Just add water",
  "format": "DVD",
  "quantity": 1,
  "condition": "like new",
  "price": 3.99,
  "languages": ["english", "spanish"],
  "actors": [ {
                "name": "Danny DeVito",
                "born": 1944
              },
              {
                "name": "Jonah Hill",
                "born": 1983
              }
            ],
  "released": 2008
}
```

```
{
  "type": "Movie",
  "title": "The Sitter",
  "format": "Blueray",
  "quantity": 1,
  "condition": "disk only",
  "price": 7.99,
  "languages": ["english", "spanish"],
  "actors": [ {
                "name": "Jonah Hill",
                "born": 1983
              },
              {
                "name": "Max Records",
                "born": 1997
              }
            ],
  "released": 2011
}
```


数据冗余!

如何做更新? 如何保证一致性?
查询路径显著增加!

过多的内联 – 实体混合


```
{
        "order": 1234,
        "orderDate": "2021-03-29T05:05:15Z",
        "customer": {
                "first": "Reva","last": "Patil",
                "address": {"city": "Allen","state": "TX",…},
                "phoneNumber": "555-123-4571",...
        },
        "lineItems": [
    {
                "type": "Movie",
                "title": "Just add water",
                "format": "DVD",
                "quantity": 1,
                "condition": "like new",
                "price": 3.99,
                "languages": ["english", "spanish"],
                "actors": [{
                        "name": "Danny DeVito",
                        "born": 1944
                }, ...
```

过多的内联 – 实体混合

PurchaseOrder

```
{
        "order": 1234,
        "orderDate": "2021-03-29T05:05:15Z",
        "customer": {
                "first": "Reva","last": "Patil",
                "address": {"city": "Allen","state": "TX",…},
                "phoneNumber": "555-123-4571",...
        },
        "lineItems": [
    {
                "type": "Movie",
                "title": "Just add water",
                "format": "DVD",
                "quantity": 1,
                "condition": "like new",
                "price": 3.99,
                "languages": ["english", "spanish"],
                "actors": [{
                        "name": "Danny DeVito",
                        "born": 1944
                }, ...
```

Customer

Product

Actor

使用链接来解决JSON中的内联

PurchaseOrder

```
{
        "order": 1234,
        "orderDate": "2021-03-29T05:05:15Z",
        "customerId":5632,
        "lineItems": [
    {
                "productId": "7637",
                "quantity": 1,
     "price":3.99
    }
  ]
}
```

Customer

```
{
  "id":5632
  "first": "Reva",
  "last": "Patil",
  "address": {
    "city": "Allen",
    "state": "TX",...},
  "phoneNumber": "555-123-4571"
}
```

Product

```
{
   "id"7637,
   "type": "Movie",
   "title": "Just add water",
   "format": "DVD",
   …
   "actors": [
      {"name": "Danny DeVito",
        "born": 1944}, …
}
```

Actor

关联, 事务性变更, 参照完整性, 等等

Ryan McGuire / Gratisography

能否把JSON和关系模型结合起来?

Demo using Autonomous JSON databas

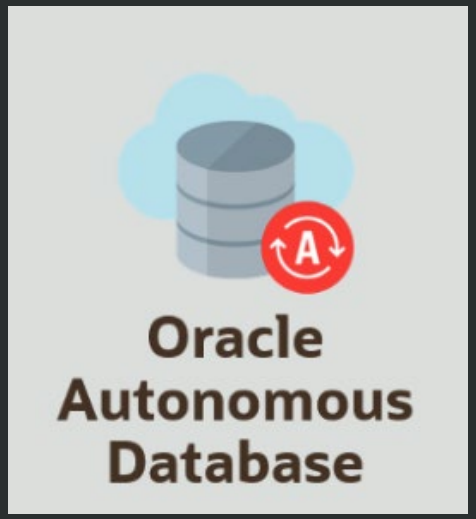
使用自治数据平台的结合二者的演示

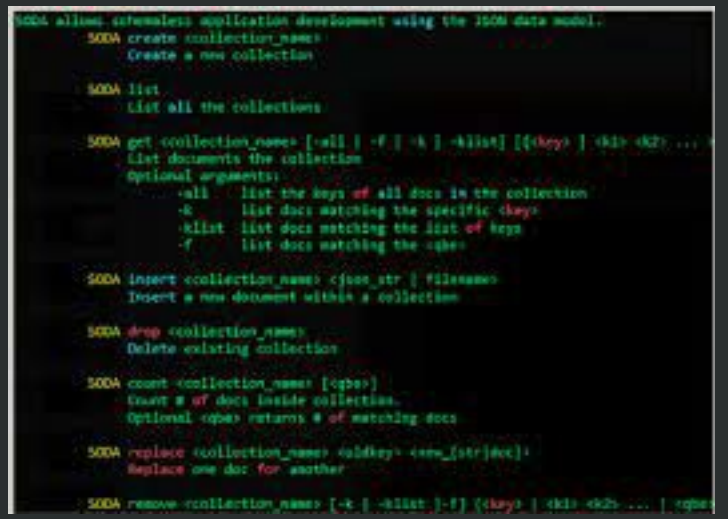
永久免费

可用且永不过期的服务

+

30天免费体验





Developing with JSON and SODA in Oracle Database Workshop

Explore JSON and SODA with the Oracle Autonomous JSON Database.

Workshop length: 1 hour, 30 minutes

Share Workshop Link

Other LiveLabs you might like

Oracle RAC Fundamentals

RESTful Services for your Autonomous Database

Modernize and Extend Legacy Apps in the Oracle Cloud

Ways to run this workshop

Choose how you want to run this workshop.

Launch **Always Free** Workshop

*More about **Always Free***

Launch **Free Trial** Workshop

*More about **Free Trial***

Run On Your **Tenancy**

*More about using Oracle Universal Credits you've purchased: **Using your credits** | **Services available***

Reserve Workshop on **LiveLabs**

*You need an Oracle account to run on the free LiveLabs tenancy: **Oracle account help** | **Oracle account signup***

Workshop Outline

Workshop Details