

پایگاه داده تمرین چهارم

عباس یزدان مهر
۹۹۲۴۳۰۷۷
مهندسی کامپیوتر، دانشگاه شهید بهشتی
abbas.yazdanmehr1@gmail.com

۱۲ دی ۱۴۰۱

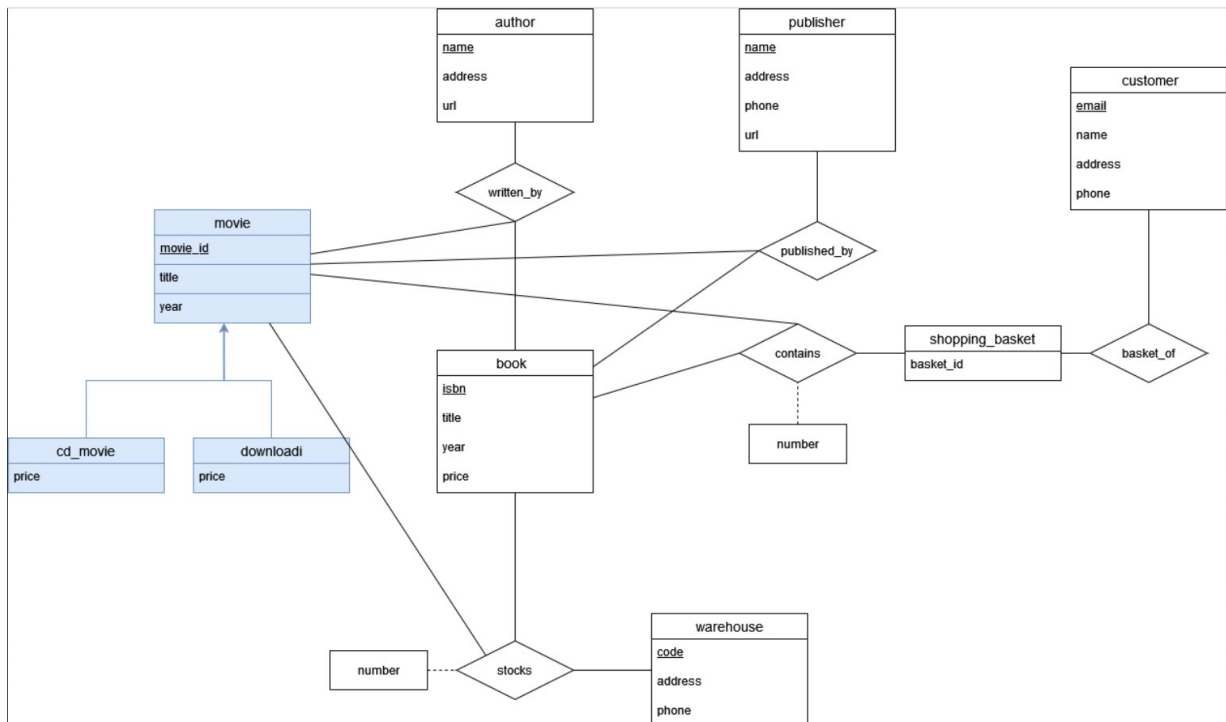
۱

الف

Entity Set	Primary Key
author(name, address, URL)	name
book(ISBN, title, year, price)	ISBN
publisher(name, address, phone, URL)	name
customer(email, name, address, phone)	email
warehouse(code)	code
shopping_basket(basket_id)	Basket_id (not explicit)

ب

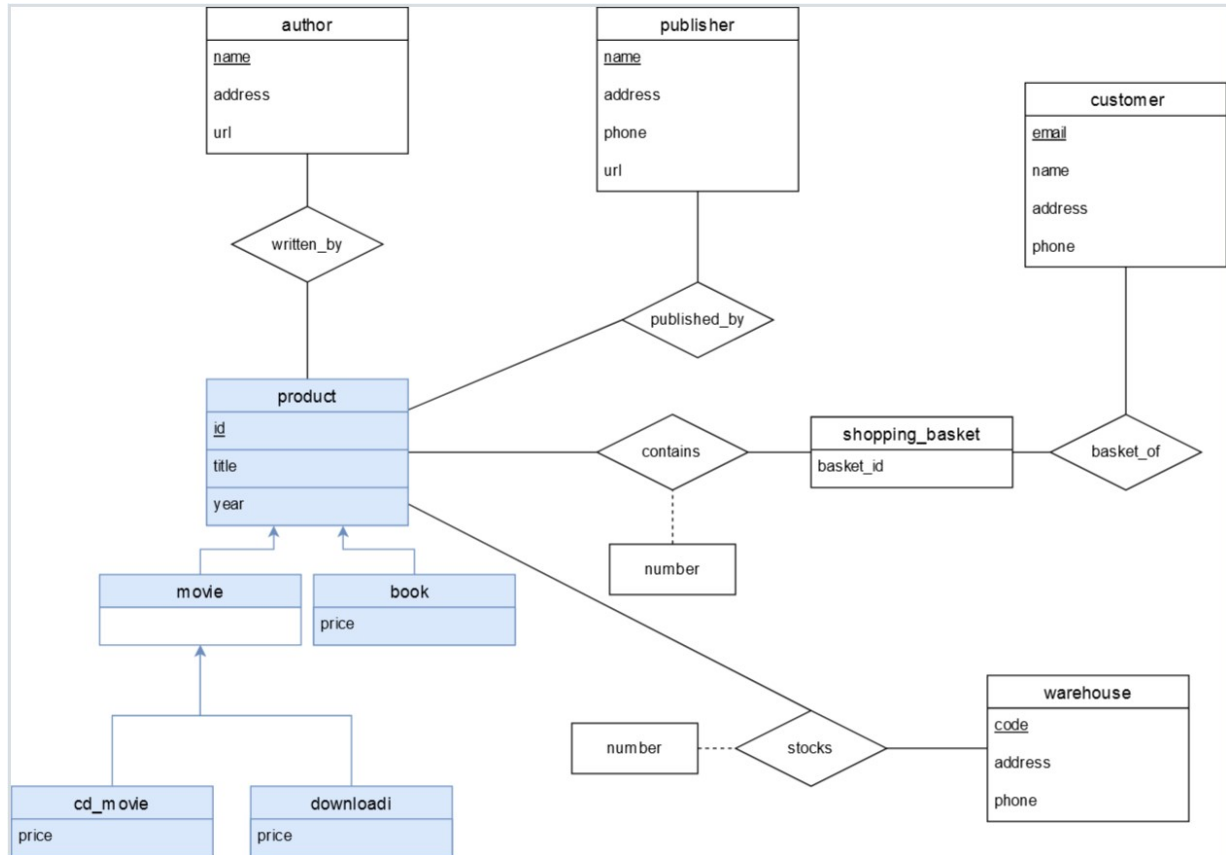
چون فیلم هم می تواند قابل دانلود باشد و هم میتواند بصورت دیسکی باشد باید از generalization overlapping استفاده کنیم.

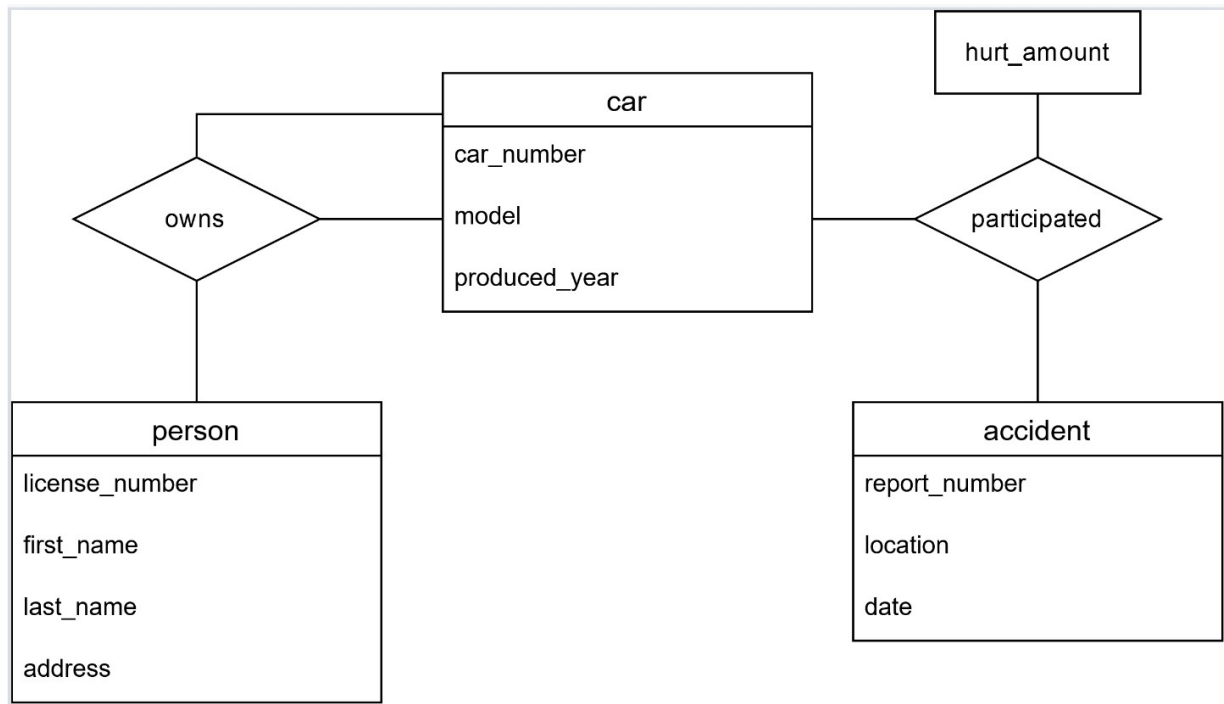


نکته

در شکل movie و book باهم در یک رابطه شرکت نمی کند و فقط یکی شرکت می کند.

ج





۳

موجودیت strong خودش حتما کلید اصلی دارد، و به دیگر موجودیت ها وابسته نیست و با یک مستطیل نمایش داده می شود، ولی موجودیت weak کلیدش را از موجودیت های قوی میگیرد که به آنها discriminator می گویند، و به آنها وابسته است و با یک مستطیل دوخطی نمایش داده میشود.

۴

۱. در پایگاه داده بهتر است هر موجودیت ویژگی های خاص خود را داشته باشد، در صورتی که فقط از strong استفاده کنیم، باید ویژگی های مربوط به موجودیت هایی که به آن وابسته ایم را هم در موجودیت وابسته بیاوریم که باعث میشود اطلاعات تکراری داشته باشیم.

۲. استفاده از موجودیت ضعیف باعث میشود که یک موجودیت به چند موجودیت وابسته شود و این باعث می شود که سرعت عملیات ها بیشتر باشد و در ضمن محدودیت هایی ایجاد می کند که باعث می شود از خطاها جلوگیری شود.

۵

شماتیک جداول:

convention: *table_name*(primary_key, foreign_key(references), simple_attribute,)

```
employee_name(name_id, fname, minit, lname)
employee(ssn, name_id, sex, address, salary, bdate,
department.name NOT NULL, department.number NOT NULL,
supervisor_ssn(employee.ssn), supervisee_ssn(employee.ssn))
project(name, number, location, department.name, department.number)
department(name, number, number_of_employee)
dependent(employee.name_id NOT NULL, sex, birth_date, relationship)
manages(ssn, department.name NOT NULL, department.number NOT NULL, start_date)
works_on(ssn NOT NULL, project.name NOT NULL, project.number NOT NULL, hours)
department_locations(department.name, department.number, location)
```

۶

همه ی این موارد مربوط به قسمت ارتباط بین سطح بالا و سطح پایین در specialization است.

همه ی این موارد جزء موارد محدود سازی ER هستند.

در صورتی که specialization به صورت disjoint باشد یک موجودیت تنها می تواند به یک موجودیت که در سطح پایین تری قرار دارد رابطه داشته باشد که یعنی هر کلاس پدر تنها می تواند به یک زیرکلاس یا کلاس فرزند مرتبط باشد و نه هر دو. مثلاً یک موجودیت پدر با نام ماشین یا میتواند مرتبط با موجودیت ماشین سنگین باشد یا موجودیت ماشین سبک و نه هر دو.

در صورتی که specialization به صورت overlapping باشد برعکس disjoint است به این صورت که هر موجودیت میتواند به یک یا چند موجودیت که زیرکلاس آن هستند مرتبط شوند. مثلاً یک موجودیت با نام شخص هم میتواند به موجودیت کارمند مرتبط شود و هم به موجودیت دانشجو.

مقایسه

در disjoint هر موجودیت سطح بالا فقط به یک موجودیت سطح پایین مرتبط می شود ولی در overlapping تعداد ارتباط با زیرکلاس ها محدودیتی ندارد.

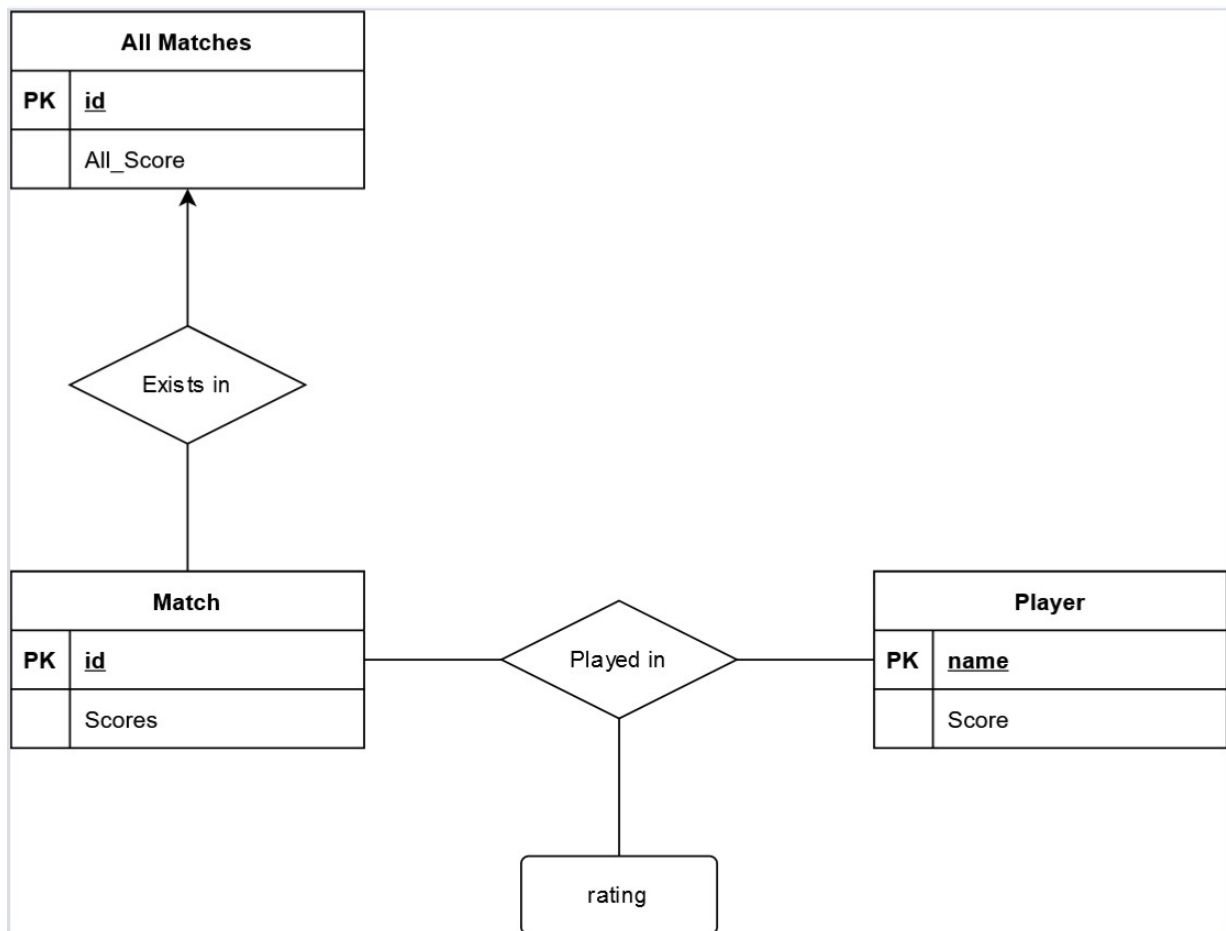
در یک total specialization سطح بالاتر حتما باید به یک موجودیت سطح پایین تر مرتبط باشد.

در یک partial specialization سطح بالا حتما نیای د به یک موجودیت سطح پایین تر مرتبط باشد بلکه میتواند ارتباطی نداشته باشد.

مقایسه

پس total specialization حتما هر موجودیت سطح بالاتر به یک موجودیت پایین تر مرتبط است اما در partial لزومی ندارد.

V



^

شماتیک جداول:

convention: *table_name*(primary_key, foreign_key(references), simple_attribute,)

student(student_id, student_name, dob, age, door_number, street, city, state, pin)
course(course_id, course_name)
subjects(subject_id, subject_name, lecturer_id, course_id)
lecturer(lecturer_id, lecturer_name, course_id, student_id)
attends(student_id, course_id)
takes(lecturer_id, course_id)
hobby(student_id, hobby)

پایان