Layout Design Deploy Help

Default Layout ×

Store primary key:
Store_key

Store
- Store_key INT
- number INT
- name VARCHAR(100)
- address VARCHAR(100)
- manager_name VARCHAR(100)
- city_name VARCHAR(100)

Sales primary keys:
1.Product
2.Payment_time
3.Order_time
4.Store
5.Client
6.Amount

Sales foreign keys:
1.Product
2.Payment_time
3.Order_time
4.Store
5.Client

Product primary key:
Product_key

Product
- Product_key INT
- number INT
- name VARCHAR(100)
- description VARCHAR(100)
- size VARCHAR(40)
- distributor VARCHAR(100)
- category VARCHAR(100)
- rate INT
- color VARCHAR(30)

digikala db schema
pouya behzadifar

Store

Sales
- Product VARCHAR(30)
- Payment_time DATETIME
- Order_time DATETIME
- Store VARCHAR(30)
- Client VARCHAR(30)
- Amount INT
- Price INT

Product

Order_time

Client

Payment_time

Client primary key:
client_key

Client
- client_key INT
- client_id INT
- client_name VARCHAR(100)
- client_address VARCHAR(100)
- client_national_code NUMERIC(0,10)
- client_post_code NUMERIC(0,12)

Time primary key:
Time_key

Time
- Time_key INT
- date DATE
- month INT
- year INT
- event VARCHAR(30)
- weekend flag BOOLEAN
- weekday flag BOOLEAN

ENG 6:19 PM 10/29/2022

۲. transaction Isolation . این خاصیت مانع هر دو تغییر برای خریدبلیت خواهد شد.

هر query که به db زده می شود، dbms چک می کند که خاصیت accuracy و consistency پایگاه داده حفظ شود. transaction isolation بیان می کند هر transaction باید تنها transaction ایی باشد که به یک resource خاص داده دسترسی پیدا می کند. در غیر این صورت جلوی آن را خواهد گرفت.

۳.

a) خاصیت null بودن یک مقدار. مثلاً ممکن است Phone-number برای یک کاربر وجود نداشته باشد (null)، ولی بعداً شماره تلفن بگیرد و نیاز باشد این مقدار به جدول اضافه شود. یا مثلاً آدرس کاربر عوض شود و ما آدرس جدید رو نداریم. در این صورت به مقدار null برای آدرس نیاز داریم.

در کل اگر یک مقدار وجود نداشته باشد یا ما مقدارش رو نداریم از null استفاده می کنیم.

b)

student

id	...	tot-cred
۱	...	۳۰۰
۲	...	۵۰۰
۳	...	۴۰۰

takes

id	...	grade
۲	...	A
۲	...	B
۳	...	C

insert : چون در جدول takes ، ID یک foreign key از جدول student است insert کردن هر instance با ID غیر از ۲ و ۳ با مشکل مواجه خواهد شد.

delete : چون در جدول takes ، ID های ۲ و ۳ استفاده شده اند، در صورتی که بخواهیم دو سه را از جدول student حذف کنیم با مشکل مواجه خواهیم شد instance با ID

۴. $\Pi_{course\_id}(\underbrace{\sigma_{semester='spring'}(section)}_{①})$

این query ، course_id همه‌ی درس‌هایی که در spring ارائه شده‌اند را از جدول section پیدا می‌دهد.

$\{CS\text{-}101, CS\text{-}190, CS\text{-}190, CS\text{-}315, CS\text{-}319\}$

* با فرض اینکه spring داخل "" است.

④ (عبارت ① − $\Pi_{course\text{-}id}(section)$) P_{α}

این query ، ابتدا تمام course_id های ROW جدول section را پیدا می‌دهد.
سپس مقادیر خروجی عبارت 1 را از آن حذف می‌کند و در جدول α ذخیره می‌کند.
در کل course_id های درس‌هایی که در spring ارائه نشده‌اند را می‌دهد.

{خروجی عبارت ①} − {CS-319, CS-347, CS-190, CS-190, CS-101, CS-101, Bio-301, Bio-101}
$= \{Bio\text{-}101, Bio\text{-}301\}$

course_id
Bio-101
Bio-301

⟶ جدول α

③ $\Pi_{y.course\_id, y.title}$ (خروجی حاصل ضرب α $\times$ ρ_{y}(course)) $\sigma_{s.course\_id = c.course\_id}$

حاصل ضرب کارتزین جدول X و جدول course با شرط این‌که course_id های دو جدول یکسان باشند.

α.course_id	y.course_id	title	dept_name	credits
Bio-101	Bio-101	Biology	Biology	4
Bio-301	Bio-301	Genetics	Biology	4

سپس select می‌کنیم title ، course_id

y.course_id	title
Bio-101	Biology
Bio-301	Genetics

a. روش اول:

$\Pi_{title, ReturnDate}(\sigma_{memberID=1409 \wedge isReturned=false \wedge Book.BookID=Borrow.BookID}(Borrow \times Book))$

روش دوم:

$\Pi_{Title, ReturnDate}(\sigma_{memberID=1409 \wedge isReturned=false}(Borrow)) \bowtie_{Book.BookID=Borrow.BookID} (Book)$

b.

$\Pi_{Name}(Member \bowtie_{member.memberID=Borrow.memberID \wedge member.categoryID=category.categoryID}$

$(Borrow \bowtie_{Borrow.BookID=Book.BookID}$

~~$Member \bowtie_{member.memberID}$~~

$(Book \bowtie_{Book.categoryID=category.categoryID \wedge categoryName="Drama"} Category))))$

c.

$\Pi_{name, Title}((Member \bowtie_{member.categoryID=Book.categoryID} Book) -$

$(Member \bowtie_{member.memberID=Borrow.memberID} Borrow \bowtie_{Book.BookID=Borrow.BookID} (Book \bowtie Borrow)))$

d.

$\Pi_{Name, Title}((Member \bowtie_{Member.memberID=Borrow.memberID}$

$(Borrow \bowtie_{Borrow.BookID=Book.BookID \wedge isReturned=false \wedge Today-returnedate>10}$

$(Book \bowtie_{Book.categoryID=category.categoryID \wedge categoryname="physics"} Category)))$

e. نام فرد و کتاب قرض گرفته شده به شرطی که جریمه دیرکرد آن حداقل با بیش از مساوی ۱۰۰۰۰۰ شده است.

f. نام کتاب های ژانر "Philosophy" که نویسندگان آن "Plato" نباشد و قرض گرفته شده ولی هنوز بازگردانده نشده است.