

گزارش

در هر بخش در مورد خروجی query توضیحاتی داده شده است.

نکته: با توجه به اینکه جدول یا رابطه ای موجود نیست که به صورت self-referential باشد، برای همین از self-join استفاده نمی شود.

:SelectionQuery1

	id	name
	5	NULL
	6	NULL
	10	NULL
	12	NULL
	14	NULL
	19	NULL
	22	NULL
	27	NULL
	28	NULL
	29	NULL
	30	NULL
	32	NULL
	34	NULL
	41	NULL
	43	NULL

معلم هایی که درسی را تاکنون بر نداشته اند و باید درسی بردارند.

:SelectionQuery2

	id	name
▶	102	c102

درس ها یا کورس هایی که معلم ندارند و باید معلمی داشته باشند که برای این query این داده را به جدول course اضافه کردیم:

102	c102	900	2020	third	26	82	dep1	build4	79	0
-----	------	-----	------	-------	----	----	------	--------	----	---

:SelectionQuery3

نشان می دهد که برای هر یک از کارشناسان آموزشی چند دانشجو با توجه به سطح تحصیلی کارشناسان موجود است و به وسیله این می فهمیم که کدام یک از کارشناسان آموزشی را باید اخراج کنیم چرا که به آن ها نیاز نداریم.

id	department	education	count
77	dep1	doctoral	0
79	dep10	master	1
37	dep100	bachelor	1
97	dep11	bachelor	0
94	dep12	doctoral	1
67	dep13	doctoral	0
20	dep14	master	0
23	dep15	bachelor	0
92	dep16	master	0
8	dep17	doctoral	0
70	dep18	doctoral	0
29	dep19	doctoral	1
65	dep2	master	1
44	dep20	master	1
11	dep21	master	0

:SelectionQuery4

شناسه و مجموع واحد دانشجویانی که کمتر از ۵ واحد دارند یا اصلا واحدی را برنداشته اند(که در این صورت مجموع واحد ها برابر null خواهد بود) و باید واحد هایی را بردارند.

id	credSum
1	3
2	4
3	NULL
4	NULL
5	NULL
6	NULL
7	NULL
8	NULL
9	NULL
10	NULL
11	NULL
12	NULL
13	NULL
14	NULL
15	NULL

توجه شود که برای این query جدول رابطه و جدول takes ساخته شده است و داده های زیر به آن اضافه شده است:

	student_id	course_id	grade
▶	1	2	0
	1	3	0
	1	5	20
	2	4	0
	2	6	20

:SelectionQuery5

شناسه و معدل دانشجویان تاکنون که کمتر از ۱۰ می باشد و برای همین باید اخراج شوند.

	id	score
▶	1	6.6667
	2	5.0000

توجه شود که نمره هر درس ضرب در تعداد واحد ها می شود و بعد درون جدول **takes** قرار می گیرد.

:SelectionQuery6

مجموع حقوق ماهانه کارکنان هر یک از دانشکده ها و می توان به کمک آن بودجه سالانه را تعیین کرد.

department	sum(monthly_salary)
dep1	1120
dep10	2583
dep100	3264
dep11	535
dep12	1698
dep13	3023
dep14	2053
dep15	2319
dep16	2473
dep17	1364
dep18	2424
dep19	1621
dep2	2331
dep20	1092

:SelectionQuery7

وسایل آزمایشگاهی که در آزمایشگاهی قرار ندارند و باید در آزمایشگاهی قرار بگیرند.

	id	name	id
▶	101	li101	NULL

توجه شود که برای این query داده های زیر به **LabInstrument** اضافه شده است:

101	ld101	900	bad	NULL
-----	-------	-----	-----	------

:SelectionQuery8

مسئولین آزمایشگاهی که در آزمایشگاهی قرار ندارند و باید در آزمایشگاهی قرار بگیرند.

id	name	id
101	ld101	NULL
102	ld102	NULL

توجه شود که برای این query داده های زیر به LabDirector اضافه شده است:

101	ld101	900	dep80	800
102	ld102	NULL	dep81	801

:SelectionQuery9

آزمایشگاه هایی که مسئول آزمایشگاه ندارند و باید مسئولی برای آن ها در نظر گرفته شود.

id	name	id
101	ld101	NULL

توجه شود که برای این query داده های زیر به Lab اضافه شده است

101	ld101	NULL
-----	-------	------

:SelectionQuery10

زمان هایی که در آن ها دو یا چندین کلاس در یک مکان، سال و ترم برگزار می شوند و باید کلاس هایی را از این زمان ها جا به جا کرد.

course_slot_id	year	semester	building	class_no	count
26	2020	third	build4	79	3

توجه شود که برای این query داده های زیر به Courses اضافه شده است:

101	c101	26	2020	third	26	82	dep1	build4	79	0
102	c102	900	2020	third	26	82	dep1	build4	79	0

:SelectionQuery11

زمان هایی که در آن ها دو یا چندین امتحان در یک مکان، سال و ترم برگزار می شوند و باید امتحان هایی را از این زمان ها جا به جا کرد.

finals_timeslot_id	year	semester	building	class_no	count
82	2020	third	build4	79	3

توجه شود که برای این query داده های زیر به Course اضافه شده است:

101	c101	26	2020	third	26	82	dep1	build4	79	0
102	c102	900	2020	third	26	82	dep1	build4	79	0

:SelectionQuery12

دانشجویانی که استاد راهنما ندارند و باید برای آن ها استاد راهنما مشخص کرد.

id	name	id
101	s101	NULL

توجه شود که برای این query داده های زیر به student اضافه شده است:

101	s101	NULL	doctoral	dep69
-----	------	------	----------	-------

:SelectionQuery13

معلم هایی که بیش از چهار دانشجو را راهنمایی می کنند و باید یک یا چندین دانشجو را به معلم های دیگر داد.

id	name	count
9	t9	5

:SelectionQuery14

دانشجویانی که پروژه ندارند و باید پروژه ای بردارند.

id	name	id
101	s101	NULL

توجه شود که برای این query داده های زیر به student اضافه شده است:

101	s101	NULL	doctoral	dep69
-----	------	------	----------	-------

:SelectionQuery15

پروژه هایی که برای آن ها استاد ناظر مشخص نشده است و باید استاد ناظری برای آن ها مشخص

شود.

id	name	id
101	p101	NULL

توجه شود که برای این query داده های زیر به project اضافه شده است:

101	p101	NULL	NULL	NULL
-----	------	------	------	------

Triggers

نکته: توجه شود رابطه های deregisters/registers عملا همان رابطه Takes می باشند و

در واقع نیازی به جدول هایی برای آن ها نمی باشد زیرا رابطه به صورت عملی می باشد و در واقع کارشناسان آموزشی یک ردیفی در جدول Takes قرار می دهند.

نکته: deleteTrigger1 و deleteTrigger2 برای رابطه takes می باشند.

نکته: برای پیاده سازی کاربران و فعال شدن trigger ای به نام UsernameTrigger از رابطه

ای به نام users استفاده می شود که هنگامی که کاربری username موجود در این رابطه را تغییر می دهد، به وسیله trigger ردیفی در UsernameChanges تولید شده و به وسیله آن کنترل می شود که کاربر در هر روز پیش از دو بار نمی تواند username خود را تغییر دهد.

نکته: توجه شود که فرض شده است که در سیستم **trigger** دیگری موجود می باشد که جلوگیری از **update** ای می کند که **id** کاربر یا **users** را تغییر می دهد.

نکته: نام کاربری هر یک از کاربران پایگاه داده به صورت اتوماتیک می باشد و در واقع همان **id** آن ها می باشد و برای همین امکان ندارد که آن را تغییر دهند. برای مثال نام کاربری پورتال همان شماره دانشجویی یا **id** می باشد که قابل تغییر نمی باشد. همچنین **id** هر یک از کاربران در یکی از جداول مربوط به خودشان و نه در همه جدول ها **unique** می باشد. برای حل این مشکل یک **entity** به نام **users** ساخته می شود که هر ردیف آن به یکی از کاربران **map** می شود و برای یکی از کاربران می باشد و شامل **username** ای می باشد که قابل تغییر هست. همچنین تغییرات در جدول **'UsernameChanges'** ذخیره می شوند. توجه شود که **field** یا ویژگی **entity** مشخص می کنید که ردیف مربوط به کدام **entity** است.