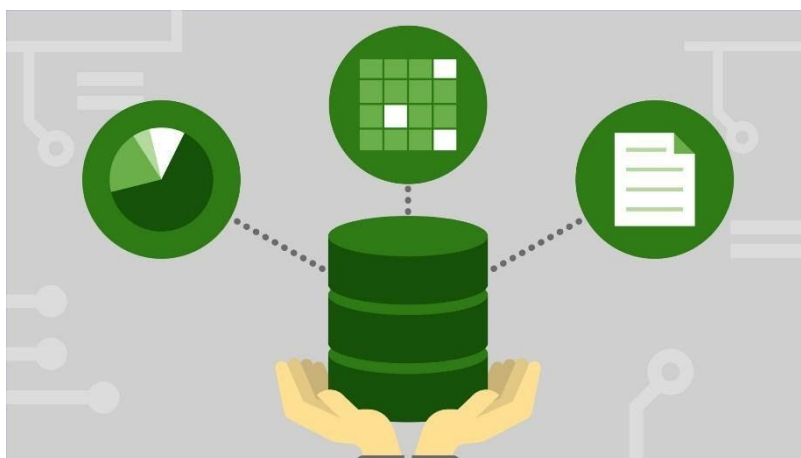


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده
دستور کار شماره ۲

شماره دانشجویی

۸۱۰۱۹۶۶۰۴

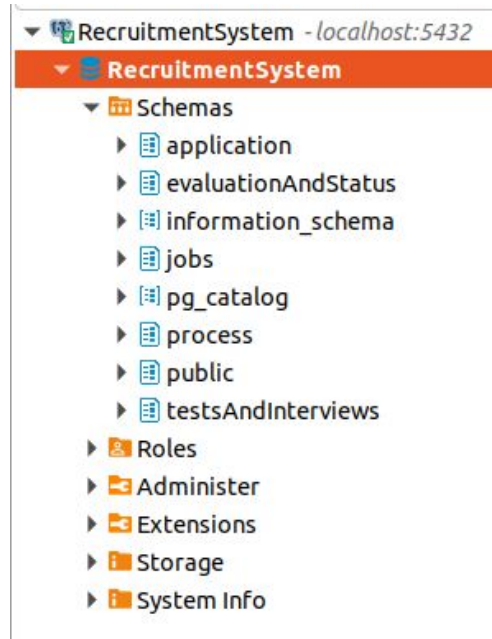
مهر ۹۹

نام و نام خانوادگی

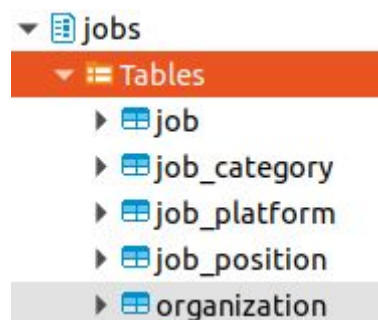
سید پارسا حسینی نژاد

گزارش فعالیت‌های انجام شده

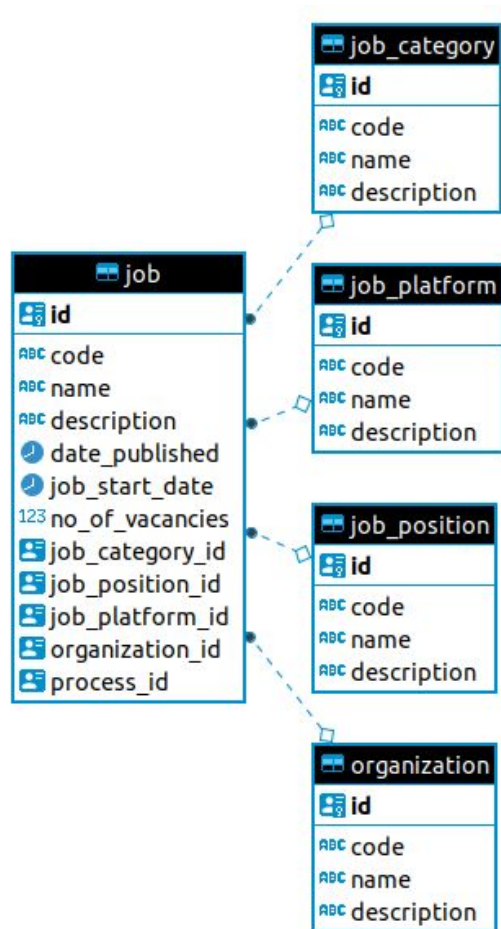
ابتدا طبق توضیحات صورت گزارش، پنج شمای مورد نظر را به وجود می‌آوریم:



سپس تک تک جداول را به همراه فیلدهایشان از طریق واسطه گرافیکی تولید می‌کنیم. دقت شود که اگر در یک جدول foreign key مربوط به جدولی دیگر داشته باشیم، باید آن جدول زودتر ایجاد شود که رعایت شد. برای مثال جداول شمای jobs را می‌بینیم:



حال برای مشاهده‌ی درست ایجاد شدن جدول، نمودار ER آن را به وسیله‌ی DBeaver می‌کشیم که همانطور که مشاهده می‌شود، درست ایجاد شده است:



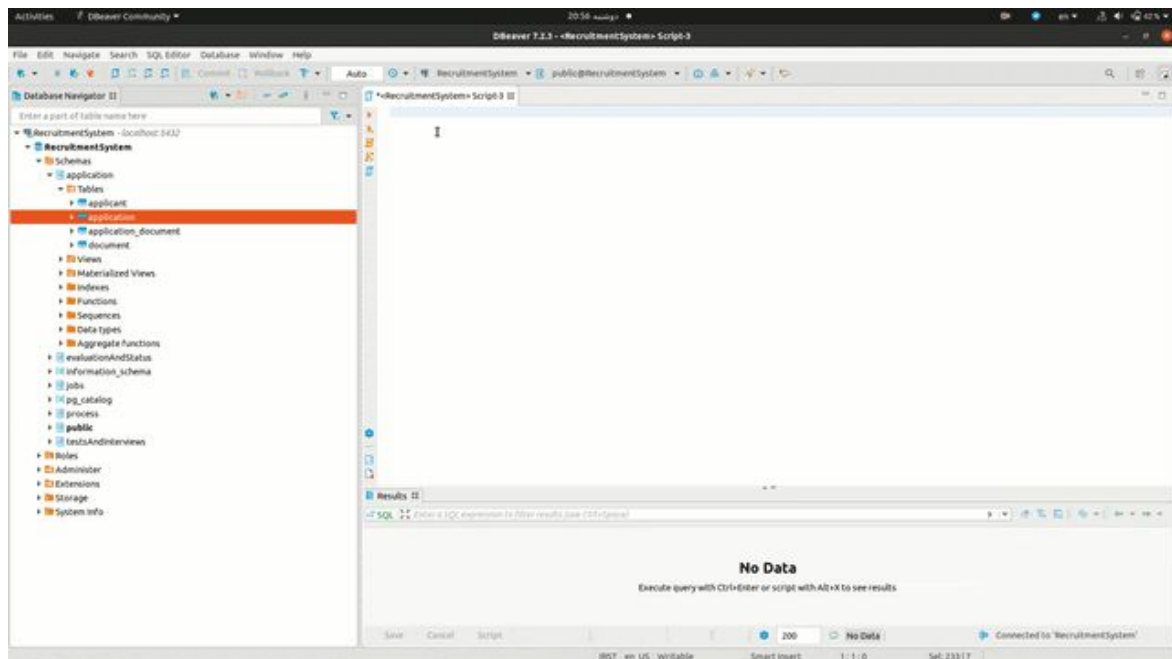
برای مثال، جدول زیر نتیجه‌ی ایجاد جدول jobs است:

	Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
Columns	id	1	uuid						<input checked="" type="checkbox"/>	uuid_generate	
Constraints	code	2	varchar	10	10			default	<input checked="" type="checkbox"/>		
Foreign Keys	name	3	varchar	100	100			default	<input checked="" type="checkbox"/>		
Indexes	description	4	varchar					default	<input checked="" type="checkbox"/>		
Dependencies	date_publish	5	timestamp		26	3			<input checked="" type="checkbox"/>		
References	job_start_dat	6	timestamp		26	3			<input checked="" type="checkbox"/>		
Partitions	no_of_vacanc	7	int4		10				<input checked="" type="checkbox"/>		
Triggers	job_category	8	uuid						<input checked="" type="checkbox"/>		
Rules	job_position	9	uuid						<input checked="" type="checkbox"/>		
Statistics	job_platform	10	uuid						<input checked="" type="checkbox"/>		
Permissions	organization	11	uuid						<input checked="" type="checkbox"/>		
DDL	process_id	12	uuid						<input checked="" type="checkbox"/>		
Virtual											

Columns	Name	Owner	Type
Constraints	PK_job	job	PRIMARY KEY
Foreign Keys			
Indexes			
Dependencies			
References			
Partitions			
Triggers			
Rules			
Statistics			
Permissions			
DDL			
Virtual			

	Name	Owner	Type	Referenced Column	Associated Entity	Comment	Match Type	Delete Rule	Update Rule
Columns									
Constraints	FK_141	job	FOREIGN KEY	PK_job_category	job_category		SIMPLE	No Action	No Action
Foreign Keys	FK_144	job	FOREIGN KEY	PK_job_position	job_position		SIMPLE	No Action	No Action
Indexes	FK_147	job	FOREIGN KEY	PK_job_platform	job_platform		SIMPLE	No Action	No Action
Dependencies	FK_150	job	FOREIGN KEY	PK_organization	organization		SIMPLE	No Action	No Action
References	FK_153	job	FOREIGN KEY	PK_process	process		SIMPLE	No Action	No Action
Partitions									
Triggers									
Rules									
Statistics									
Permissions									
DDL									
Virtual									

حال با استفاده از کنسول DBeaver، یک جدول تست می‌سازیم:



```
create table application.test(
  id uuid not null default uuid_generate_v4(),
  test_field varchar(100) not null,
  test_pk uuid not null,
  constraint "test_pk" primary key (id),
  constraint "test_fk" foreign key (test_pk) references application.application(id)
)
```

حال با استفاده از ابزار گرافیکی، در هر جدول سه رکورد اضافه می‌کنیم. ابتدا از جدول recruiter شروع کرده و سپس به ترتیب شما های موجود در سایت مذکور، جداول را پر می‌کنیم (برای این که جدول process به این جدول نیاز دارد).

recruiter Enter a SQL expression to filter results (use Ctrl+Space)			
Grid	id	first_name	last_name
1	c2b5e44d-ff85-4784-9d11-b96ef9af3eb6	a1	a2
2	[NULL]	b1	b2
3	[NULL]	c1	c2
Text			

حال با استفاده از کنسول، سه داده در یک جدول insert می‌کنیم.

```
INSERT INTO "evaluationAndStatus".application_status_change
(id, date_changed, application_status_id, application_id)
VALUES
(uuid_generate_v4(), '2016-06-22 19:10:25', '7f3b597b-24d6-4c8a-817b-aacee8c5f0b9', '9009b2f5-89c3-40f4-9834-b8d22878df57'),
(uuid_generate_v4(), '2017-06-22 19:10:25', '7f3b597b-24d6-4c8a-817b-aacee8c5f0b9', '9009b2f5-89c3-40f4-9834-b8d22878df57'),
(uuid_generate_v4(), '2018-06-22 19:10:25', '7f3b597b-24d6-4c8a-817b-aacee8c5f0b9', '9009b2f5-89c3-40f4-9834-b8d22878df57');
```

حال با استفاده از دستور ALTER TABLE، یک ستون از جدولی که ابتدا به صورت تست ایجاد کرده بودیم حذف می‌کنیم.

Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
id	1	uuid						<input checked="" type="checkbox"/>	uuid_generate	
test_field	2	varchar	100	100			default	<input checked="" type="checkbox"/>		
test_pk	3	uuid						<input checked="" type="checkbox"/>		

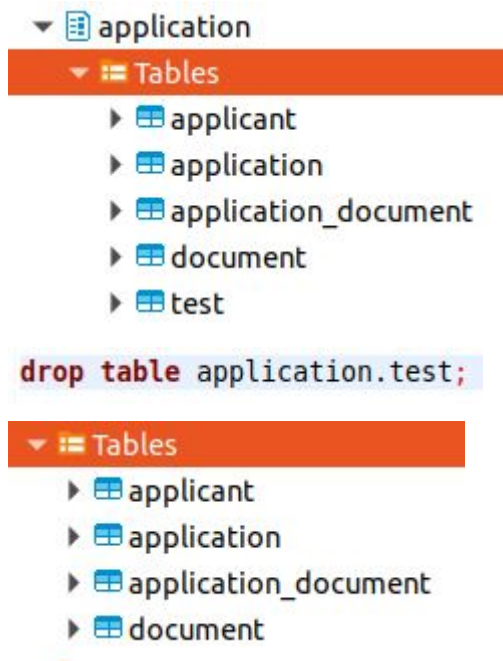
```
ALTER TABLE application.test
DROP COLUMN test_field;
```

Column Name	#	Data type	Length	Precision	Scale	Identity	Collation	Not Null	Default	Comment
id	1	uuid						<input checked="" type="checkbox"/>	uuid_generate	
test_pk	3	uuid						<input checked="" type="checkbox"/>		

حال با استفاده از دستور ALTER TABLE، یک ستون دیگر به جدولی که ابتدا به صورت تست ایجاد کرده بودیم اضافه می‌کنیم.

```
ALTER TABLE application.test
ADD COLUMN new_test_field varchar(50);
```

حال این جدول را با استفاده از دستور DROP حذف می‌کنیم.



حال سعی می‌کنیم یک داده از جدول recruiter که به آن وابستگی وجود دارد حذف کنیم. مطابق انتظار، این سطر حذف نخواهد شد چون جداول دیگر به آن داده foreign key دارند و با error مواجه خواهیم شد.



حال وارد قسمت query ها می‌شویم:

(الف)

```
select distinct "name"
from jobs.job_category
```

name
job_cat1
job_cat2
job_cat3

ب) از group by برای کنار هم گذاشتن شغل‌های با نوع یکسان استفاده می‌کنیم تا بتوانیم تعداد هر یک را حساب کنیم.

```
select count(*)
from "evaluationAndStatus".applicant_evaluation ae, application.application a, jobs.job j
where ae.application_id = a.id and a.jobs_id = j.id
group by j.job_category_id;
```

count
3

پ) قسمت اول:

```
select distinct jp."name"
from application.applicant a, jobs.job j, jobs.job_position jp, application.application a2
where a.id = a2.applicant_id and a2.jobs_id = j.id and j.job_position_id = jp.id;
```

name
job_position1

قسمت دوم: از عبارت not in استفاده می‌کنیم و به وسیله‌ی آن تمام کارجویانی که تقاضای کار کرده‌اند را پیدا کرده و چک می‌کنیم که این کارجو در آن لیست نباشد.


```
select a.*
from application.applicant a
where a.id not in (select a2.applicant_id
                  from application.applicant a2);
```

applicant

select a.* from application.applicant a where

id	first_name	last_name	email	phone	summary
954449a2-413a-4982-839d-3cc525dfc4fd	applicant2	applicant2	applicant2@gmail.com	09121231212	summ2
f0fe2a2b-3621-4e56-a9b8-7474a2c41f02	applicant3	applicant3	applicant3@gmail.com	09121231212	summ3

ت) ابتدا با استفاده از application_id کد jobs_id را استخراج می‌کنیم و با استفاده از این کد process_id را پیدا می‌کنیم. حال به ازای این شناسه، تمامی process step های موجود را پیدا کرده و چک می‌کنیم که status آن ها done باشد.

ث) از Left outer join استفاده می‌کنیم تا اگر به ازای یک job، سازمان مربوطه null بود، باز هم جایگاه شغلی را نشان دهد. حال با استفاده از job_id، موقعیت مربوطه را پیدا کرده و هر دو را نشان می‌دهیم.

```
SELECT jp."name", o."name"
FROM jobs.job_position jp, jobs.organization o
LEFT JOIN jobs.job j on o.id = j.organization_id
where jp.id = j.job_position_id;
```

job_position(+)

SELECT jp."name", o."name" FROM jobs.job_p

	name	name
1	job_position1	job_organization1
2	job_position1	job_organization1
3	job_position1	job_organization1

ج) برای چک کردن وجود داشتن مدارک این را چک می‌کنیم که به ازای متقاضی، document != null

```
select a2.*
from "testsAndInterviews".interview i, application.application a, application.applicant a2, application."document" d, application.application_document ad
where i.application_id = a.id and a.jobs_id = '67834df2-9dc7-4d09-a41e-6512579478fc'
and a2.id = a.applicant_id and ad.application_id = a.id and ad.document_id = d.id and d."document" notnull;
```

applicant

select a2.* from "testsAndInterviews".interview

id	first_name	last_name	email	phone	summary	Value
----	------------	-----------	-------	-------	---------	-------

چ) قسمت اول:

```
select j."name" , jp."name"
from jobs.job j , jobs.job j2 , jobs.job_position jp
where j."name" = 'job1'
and j."name" = j2."name"
and j2.job_position_id = jp.id;
```

job(+) ☒

select j."name", jp."name" from jobs.job j , jo | Enter a SQL expres

ABC name	ABC name
job1	job_position1

قسمت دوم: از group by برای کنار هم گذاشتن شغل‌های با نام یکسان استفاده کرده و سپس با استفاده از having آن‌هایی را که تعداد کمتر مساوی یک دارند کنار می‌گذاریم.

```
select j."name"
from jobs.job j
group by j."name"
having count(j.job_position_id) > 1;
```

job ☒

select j."name" from jobs.job j group by j."nan | Ent

ABC name

(ح)

```
select t.max_score, a3.answer_details, a3.total_grades, in2.notes
from application.applicant a, application.application a2, "testsAndInterviews".application test at2,
"testsAndInterviews".test t, "testsAndInterviews".answers a3, "testsAndInterviews".interview i, "testsAndInterviews".interview_note in2
where at2.application_id = a2.id and a.id = a2.applicant_id and t.id = at2.test_id and a3.application_test_id = at2.id and i.application_id = a.id
and in2.interview_id = i.id
```

test(+) ☒

select t.max_score, a3.answer_details, a3.total | Enter a SQL expression to filter results (use Ctrl+Space)

max_score	answer_details	total_grades	notes
123			

Value ☒

(خ) بله. چون جدول process، هیچ رابطه‌ی foreign key ندارد که به واسطه‌ی آن به جدول شغل مرتبط شود پس می‌توان از این داده برای شغل‌های مختلف استفاده کرد.

(د) از group by استفاده می‌کنیم تا بتوانیم به ازای platform های مختلف، تعداد هر یک را محاسبه کنیم.

```
select job_platform_id , count(*)  
from jobs.job j2  
group by j2.job_platform_id
```

job_platform_id	count
9788769b-dafc-4940-b14c-4d4d2970efad	3