

# PostgreSQL

اسکرین‌شات مربوط به هر بخش را با اسم مرتبط داخل یک فایل zip قرار داده و اپلود کنید.

## بخش ۱: اجرای PostgreSQL

۱. بررسی کنید که **Docker** روی سیستم شما نصب شده است. در صورت نیاز آن را روی سیستم خود نصب کنید.

۲. یک کانتینر **PostgreSQL** اجرا کنید:

```
1 | docker run --name pg_container -e POSTGRES_USER=admin -e POSTGRES
```

شما می‌توانید در این مرحله Username و Password دلخواه برای یوزر Root خود ست کنید.  
۳. اسکرین‌شات بگیرید.

## بخش ۲: اجرای شل PostgreSQL

۱. با استفاده از دستور زیر وارد کانتینر دیتابیس خود شوید:

```
1 | docker exec -it pg_container bash
```

۲. با استفاده از دستور زیر به PostgreSQL وارد شوید:

```
1 | psql -U postgres
```

۳. با استفاده از دستور مناسب لیست پایگاه‌های داده موجود را نمایش دهید.  
۴. اسکرین‌شات بگیرید.

## بخش ۳: ایجاد چند دیتابیس و کاربر

۱. با استفاده از دستور مناسب، سه دیتابیس با نام‌های `blog` و `company` و `university` ایجاد کنید.

۲. با استفاده از دستور مناسب، سه کاربر با نام و پسورد دلخواه ایجاد کنید.

۳. دسترسی‌ها را طوری تعیین کنید که هر کدام از کاربران بالا به یکی از دیتابیس‌های فوق دسترسی داشته باشند.

۴. لیست همه کاربران را نشان دهید.

۵. لیست همه دیتابیس‌ها را نشان دهید.

۶. اسکرین‌شات بگیرید.

## بخش ۴: ایجاد جدول و درج داده‌ها

۱. به پایگاه داده university متصل شوید.

۲. یک Table به نام Students ایجاد کنید:

```
1 CREATE TABLE students (  
2   id SERIAL PRIMARY KEY,  
3   name VARCHAR(50),  
4   age INT,  
5   major VARCHAR(50)  
6 );
```

۳. با استفاده از دستور مناسب، تعدادی داده وارد این Table کنید.

۴. کوئری بزنید و داده‌های وارد شده در این Table را مشاهده کنید.

۵. اسکرین‌شات بگیرید.

## بخش ۵: گرفتن بکاپ از کل DBMS

۱. ابتدا از PostgreSQL و کانتینر داکر خارج شوید.

```
1 \q  
2 exit
```

۲. سپس با استفاده از دستور مناسب، یک فایل dump از کل دیتابیس بگیرید و در سیستم خود ذخیره کنید.

۳. اسکرین‌شات بگیرید.

## مرحله ۶: حذف کانتینر و راه‌اندازی مجدد PostgreSQL

۱. کانتینر مربوطه را حذف کنید.

- 1 | `docker stop pg_container`
- 2 | `docker rm pg_container`

**اگر برای دیتابیس خود docker volume تعریف کرده‌اید نیاز است آن را هم حذف کنید!**

۲. یک کانتینر جدید PostgreSQL به نام `pg_new_container` اجرا کنید.
۳. وارد کانتینر جدید شوید.
۴. لیست کاربران و دیتابیس‌ها را مشاهده کنید و مطمئن شوید که حذف شده‌اند.
۵. اسکرین‌شات بگیرید.

## **مرحله ۶: بازیابی کل پایگاه داده**

۱. از کانتینر خارج شوید.
۲. با استفاده از دستور مناسب، فایل `dump` را بازیابی کنید.
۳. وارد کانتینر و PostgreSQL شوید، سپس لیست کاربران و دیتابیس‌ها را مشاهده کنید.
۴. داده‌های موجود در جدول `Students` را مشاهده کنید.
۵. اسکرین‌شات بگیرید.

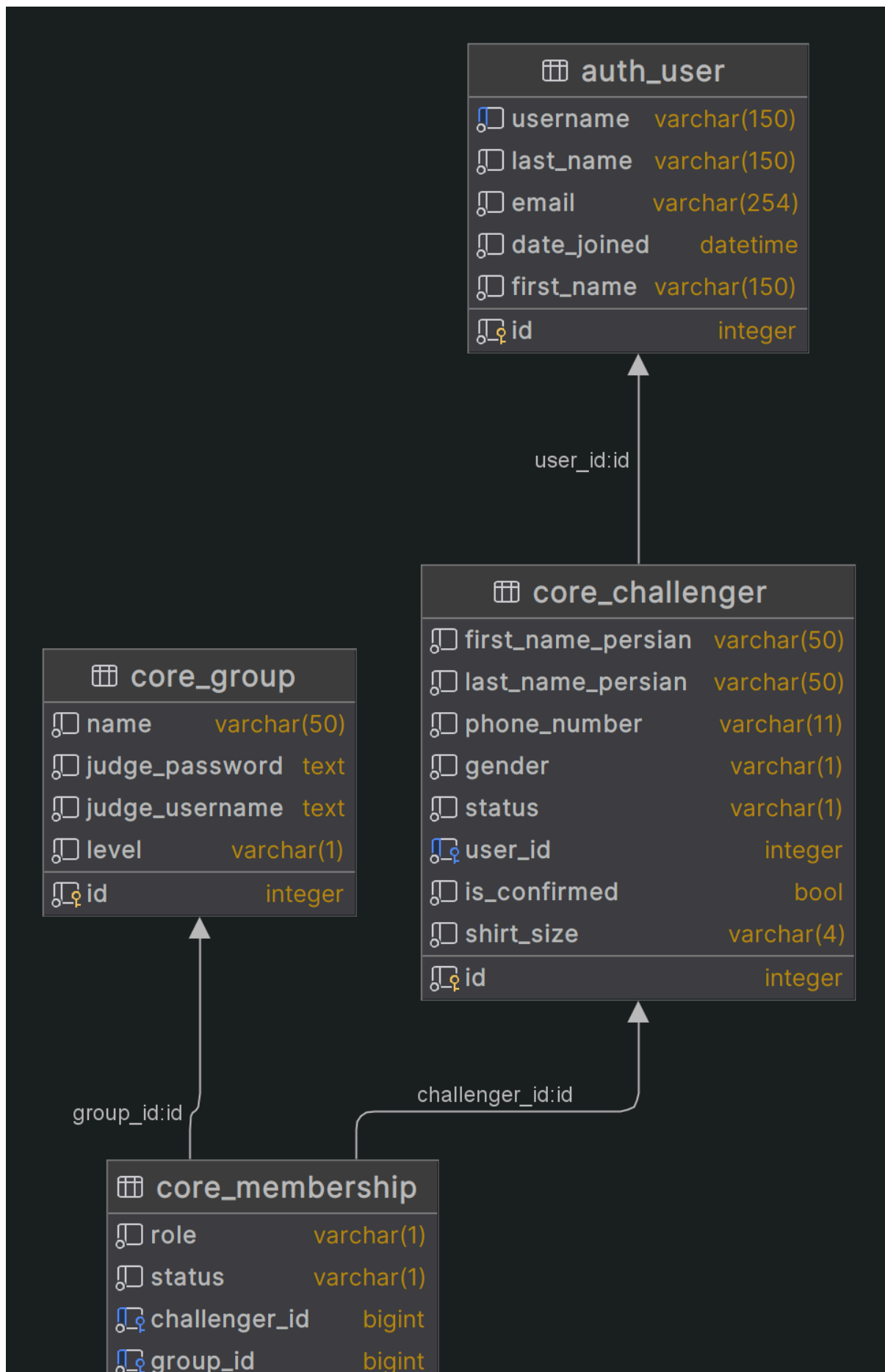
## Codocodile

کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های `part1.sql` و `part2.sql` و ... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در `root` فایل zip قرار بگیرند.

### بخش ۱: ساخت جداول

با توجه به نمودار ERD زیر، دستورهای DDL مربوط به ساخت جداول در PostgreSQL را بنویسید. نیاز است دستور را طوری بنویسید که اگر جداول موجود هستند مجدد ایجاد نشوند.

- تایپ داده‌های موجود در هر جدول را دقیقاً مشابه به نمودار ست کنید.
- نیاز است که تمامی `Primary Key` ها و `Foreign Key` های جداول ست شوند.
- نیاز است ترتیب ایجاد جداول طوری باشد که `Foreign Key` ها به درستی ست شوند.



 id

integer

## IFSC

کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های `part1.sql` و `part2.sql` و ... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در `root` فایل zip قرار بگیرند.

شما می‌توانید از این دیتابیس برای تست کوئری‌های خود استفاده کنید.

ساختار جداول این دیتابیس به این صورت است:

events	
name	text
start_date	text
end_date	text
location	text
event_id	integer

athletes	
firstname	text
lastname	text
country	text
birthday	text
gender	text
paraclimbing_sport_class	text
height	integer
speed_personal_best_score	real
speed_personal_best_date	text
speed_personal_best_round	text
athlete_id	integer

entries	
discipline	text
category	text
rank	integer
qualification_rank	integer
qualification_score	real
semi_final_rank	integer
semi_final_score	real
final_rank	integer
final_score	real
event_id	integer
athlete_id	integer

event\_id

athlete\_id



به کوثری‌های زیر با توجه به ساختار پاسخ ارائه شده جواب دهید:

۱- ورزشکاری که بیشترین تعداد مقام اول را در تمامی رویدادها کسب کرده است، پیدا کنید.

athlete_id	lastname	gold_medals_count
1227	MOEEIN	22

۲- ورزشکاری که بالاترین امتیاز مرحله مقدماتی را در هر رویداد کسب کرده است، پیدا کنید.

event_name	lastname
IFSC - Climbing World Cup 2020	ROGORA

۳- کشوری که بیشترین تعداد قهرمانی در رویدادها را دارد، پیدا کنید.

country	total_wins
IR	85

۴- تمامی ورزشکارانی که در چندین رشته مختلف مقام اول را کسب کرده‌اند، پیدا کنید.

athlete_id	firstname	lastname	unique_disciplines
68	Dinara	FAKHRITDINOVA	2

۵- جوان‌ترین ورزشکاری که تاکنون در یک رویداد برنده شده است و کمترین id را دارد، پیدا کنید.

athlete_id	firstname	lastname	youngest_age
25	Sergei	LUZHETSKII	24

توجه: پاسخ‌های ارائه شده در هر بخش درست نیستند و صرفاً ساختار جواب درست را نشان می‌دهند.

## History of Baseball

کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های `part1.sql` و `part2.sql` و ... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در root فایل zip قرار بگیرند. در این سوال به جای فایل Dump یک دیتابیس Postgres از یک دیتابیس Sqlite استفاده شده تا شما نحوه اتصال و کار با آن را بیاموزید.

شما می‌توانید از این دیتابیس [SQLite](#) برای تست کوئری‌های خود استفاده کنید.

ساختار جداول این دیتابیس به این صورت است:

player_award_vote	
award_id	text
year	integer
league_id	text
player_id	text
points_won	numeric
points_max	integer
votes_first	numeric

park	
park_id	text
park_name	text
park_alias	text
city	text
state	text
country	text

college	
college_id	text
name_full	text
city	text
state	text
country	text

fielding_outfield	
player_id	text
year	integer
stint	integer
gif	numeric
gcf	numeric
grf	numeric

manager_award	
player_id	text
award_id	text
year	integer
league_id	text
tie	text
notes	numeric

player_award	
player_id	text
award_id	text
year	integer
league_id	text
tie	text
notes	text

team_franchise	
franchise_id	text
franchise_name	text
active	text
na_assoc	text

player_college	
player_id	text
college_id	text
year	integer

salary	
year	integer
team_id	text
league_id	text
player_id	text
salary	integer

team_half	
year	integer
league_id	text
team_id	text
half	integer
div_id	text
div_win	text
rank	integer
g	integer
w	integer
l	integer

fielding	
player_id	text
year	integer
stint	integer
team_id	text
league_id	text
pos	text
g	integer
gs	numeric
inn_outs	numeric
po	numeric
a	numeric
e	numeric
dp	numeric
pb	numeric
wp	numeric
sb	numeric
cs	numeric
zr	numeric

manager	
player_id	text
year	integer
team_id	text
league_id	text
inseason	integer
g	integer
w	integer
l	integer
rank	numeric
plyr_mgr	text

hall_of_fame	
player_id	text
yearid	integer
votedby	text
ballots	numeric
needed	numeric
votes	numeric
inducted	text
category	text
needed_note	text

home_game	
year	integer
league_id	text
team_id	text
park_id	text
span_first	text
span_last	text
games	integer
openings	integer
attendance	integer

postseason	
year	integer
round	text
team_id_winner	text
league_id_winner	text
team_id_loser	text
league_id_loser	text
wins	integer
losses	integer
ties	integer

all_star	
player_id	text
year	integer
game_num	integer
game_id	text
team_id	text
league_id	text
gp	numeric
starting_pos	numeric

manager_half	
player_id	text
year	integer
team_id	text
league_id	text
inseason	integer
half	integer
g	integer
w	integer
l	integer
rank	integer

fielding_postseason	
player_id	text
year	integer
team_id	text
league_id	text
round	text
pos	text
g	integer
gs	numeric
inn_outs	numeric
po	integer
a	integer
e	integer
dp	integer
tp	integer
pb	numeric
sb	numeric
cs	numeric

pitching_postseason	
player_id	text
year	integer
round	text
team_id	text
league_id	text
w	integer
l	integer
g	integer
gs	integer
cg	integer
sho	integer
sv	integer
ipouts	integer
h	integer
er	integer
hr	integer
bb	integer
so	integer
baopp	text
era	numeric
ibb	numeric
wp	numeric

manager_award_vote	
award_id	text
year	integer
league_id	text
player_id	text
points_won	integer
points_max	integer
votes_first	integer

batting_postseason	
year	integer
round	text
player_id	text
team_id	text
league_id	text
g	integer
ab	integer
r	integer
h	integer
double	integer
triple	integer
hr	integer
rbi	integer
sb	integer
cs	numeric
bb	integer
so	integer
ibb	numeric

batting	
player_id	text
year	integer
stint	integer
team_id	text
league_id	text
g	integer
ab	numeric
r	numeric
h	numeric
double	numeric
triple	numeric
hr	numeric
rbi	numeric
sb	numeric
cs	numeric
bb	numeric
so	numeric
ibb	numeric

	hbp	numeric		hbp	numeric		hbp	numeric
	bk	numeric		sh	numeric		sh	numeric
	bfp	numeric		sf	numeric		sf	numeric
	gf	integer		g_idp	numeric		g_idp	numeric
	r	integer						
	sh	numeric						
	sf	numeric						
	g_idp	numeric						

team	
year	integer
league_id	text
team_id	text
franchise_id	text
div_id	text
rank	integer
g	integer
ghome	numeric
w	integer
l	integer
div_win	text
wc_win	text
lg_win	text
ws_win	text
r	integer
ab	integer
h	integer
double	integer
triple	integer
hr	integer
bb	integer
so	numeric
sb	numeric
cs	numeric
hbp	numeric
sf	numeric
ra	integer
er	integer
era	numeric
cg	integer
sho	integer
sv	integer
ipouts	integer
ha	integer
hra	integer
bba	integer
soa	integer
e	integer
dp	numeric
fp	numeric
name	text
park	text
attendance	numeric
bpf	integer
ppf	integer
team_id_br	text
team_id_lahman45	text
team_id_retro	text

appearances	
year	integer
team_id	text
league_id	text
player_id	text
g_all	numeric
gs	numeric
g_batting	integer
g_defense	numeric
g_p	integer
g_c	integer
g_1b	integer
g_2b	integer
g_3b	integer
g_ss	integer
g_lf	integer
g_cf	integer
g_rf	integer
g_of	integer
g_dh	numeric
g_ph	numeric
g_pr	numeric

pitching	
player_id	text
year	integer
stint	integer
team_id	text
league_id	text
w	integer
l	integer
g	integer
gs	integer
cg	integer
sho	integer
sv	integer
ipouts	numeric
h	integer
er	integer
hr	integer
bb	integer
so	integer
baopp	numeric
era	numeric
ibb	numeric
wp	numeric
hbp	numeric
bk	integer
bfp	numeric
gf	numeric
r	integer
sh	numeric
sf	numeric
g_idp	numeric

player	
player_id	text
birth_year	numeric
birth_month	numeric
birth_day	numeric
birth_country	text
birth_state	text
birth_city	text
death_year	numeric
death_month	numeric
death_day	numeric
death_country	text
death_state	text
death_city	text
name_first	text
name_last	text
name_given	text
weight	numeric
height	numeric
bats	text
throws	text
debut	text
final_game	text
retro_id	text
bbref_id	text

به کوثری‌های زیر با توجه به ساختارهای داده شده جواب دهید:

۱- به ازای هر سال، لیگ و رده، برترین تیم را مشخص کنید. نتیجه را برحسب سال به صورت صعودی مرتب کنید.

year	div_id	league_id	team_id	name
------	--------	-----------	---------	------

۲- اسامی بازیکنان برترین تیم را مشخص کنید.

year	div_id	league_id	team_id	name	player_id	name_first	name_last
------	--------	-----------	---------	------	-----------	------------	-----------

۳- جوایز این بازیکنان را لیست کنید.

year	div_id	league_id	team_id	name	player_id	award_id
------	--------	-----------	---------	------	-----------	----------

۴- درآمد این بازیکنان را نمایش دهید.

year	div_id	league_id	team_id	name	player_id	salary
------	--------	-----------	---------	------	-----------	--------

۵- نام و ایالت کالج این بازیکنان را مشخص کنید.

year	div_id	league_id	team_id	name	player_id	name_full	state
------	--------	-----------	---------	------	-----------	-----------	-------

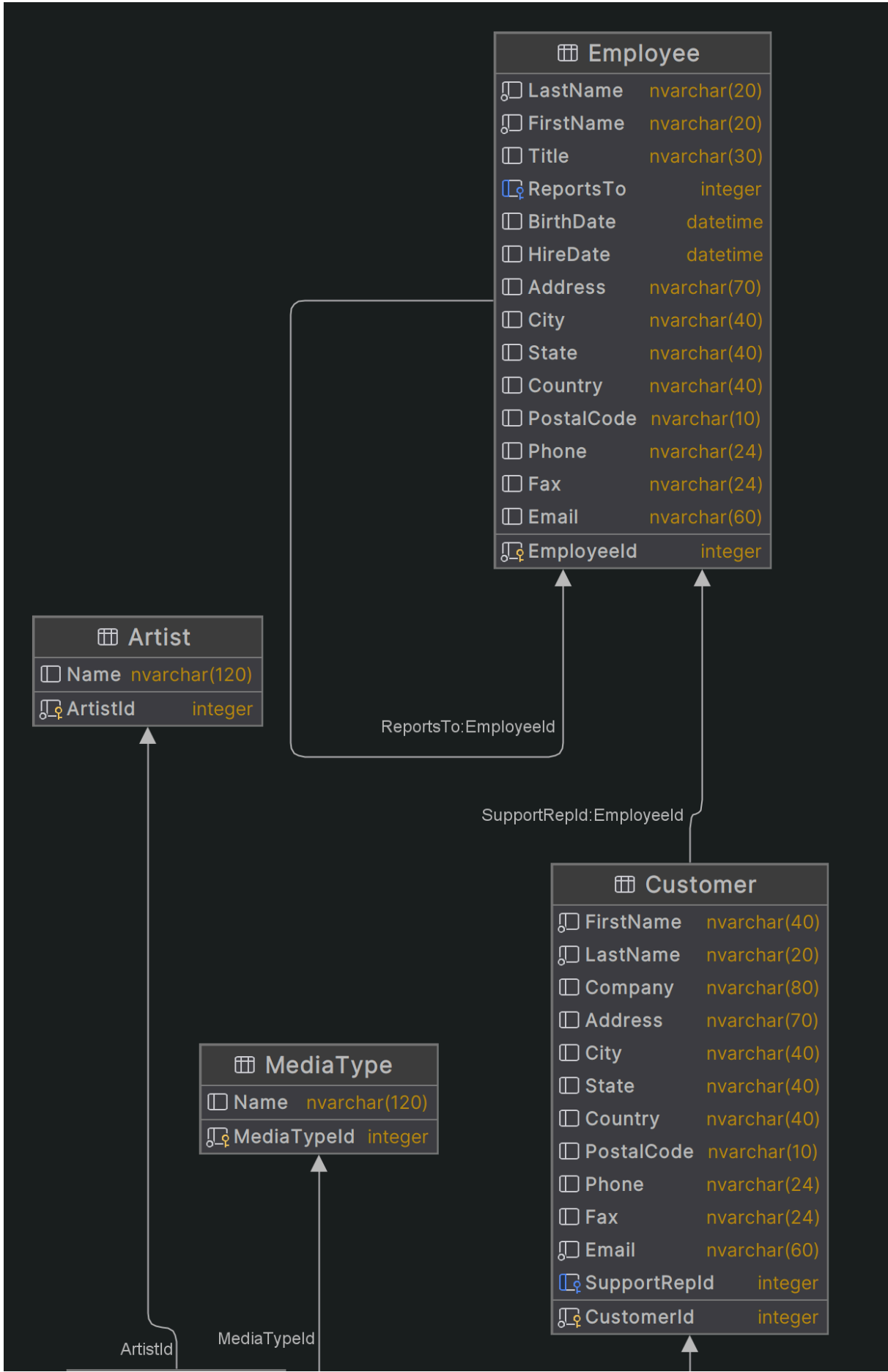
## Chinook

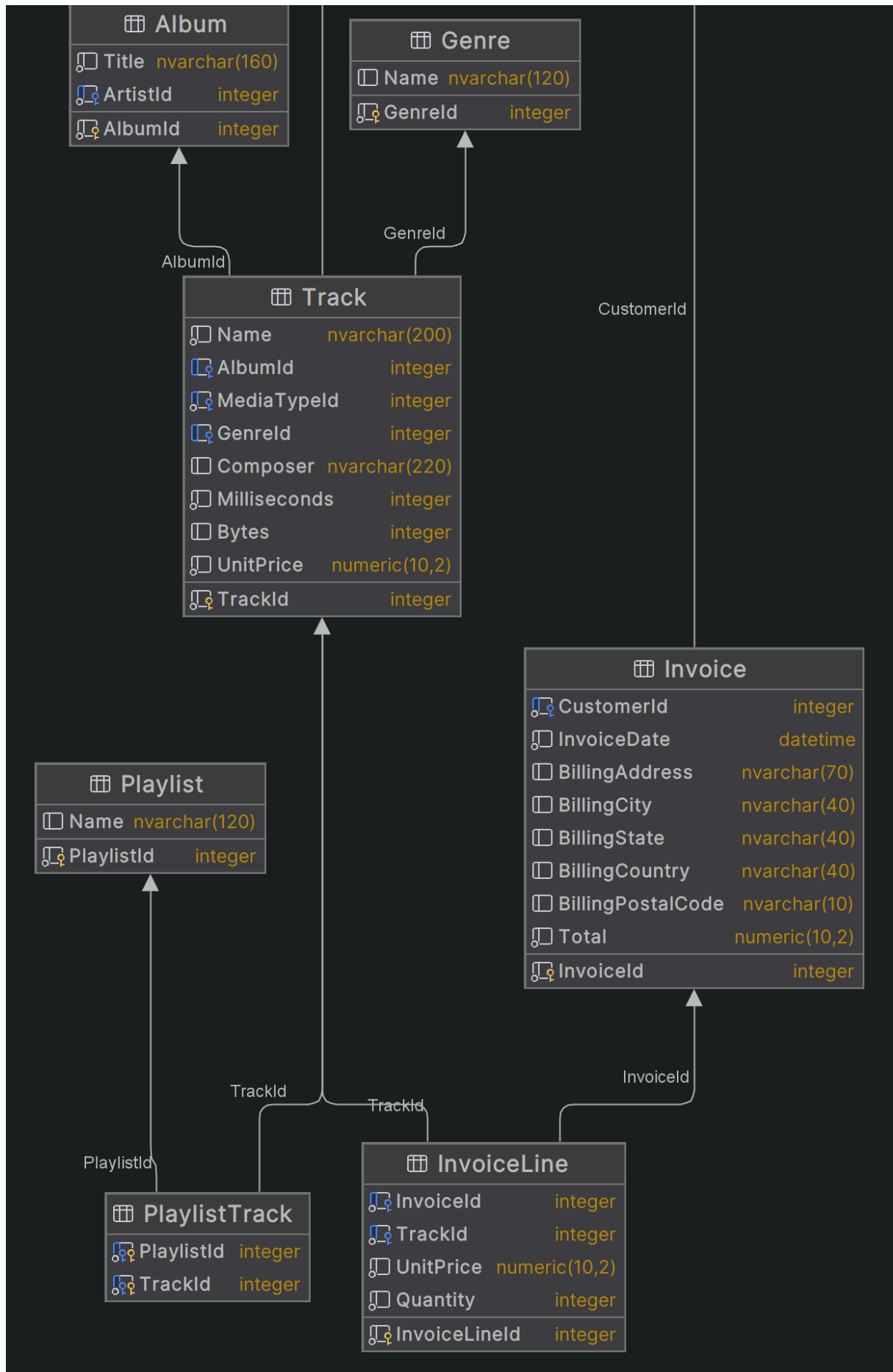
کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های `part1.sql` و `part2.sql` و... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در `root` فایل zip قرار بگیرند.

یک شرکت فروش آثار موسیقی قصد دارد جهت بررسی عملکرد و همچنین بازنگری در سیاست‌های مدیریتی و فروش خود گزارشی داده‌محور از روی پایگاه‌داده شرکت تهیه کند. بخش‌هایی از این گزارش به شما سپرده شده است.

شما می‌توانید از این دیتابیس برای تست کوئری‌های خود استفاده کنید.

ساختار جداول این دیتابیس به این صورت است:







به کوئری‌های زیر جواب دهید:

۱- مشتریانی را شناسایی کنید که در آهنگ‌های مربوط به ژانر مورد علاقه‌شان (ژانری که بیشترین تعداد خرید را از آن انجام داده‌اند) هزینه بیشتری نسبت به سایر ژانرها صرف کرده‌اند. برای هر یک از این مشتریان، شناسه، نام، نام خانوادگی، ژانر مورد علاقه و مجموع هزینه‌ای که در آن ژانر صرف شده را فهرست کنید. نتایج را بر اساس مجموع هزینه صرف شده در ژانر مورد علاقه به ترتیب نزولی مرتب نمایید.

CustomerId	FirstName	LastName	favorite_genre	total_spent_on_favorite
------------	-----------	----------	----------------	-------------------------

۲- آهنگ‌هایی را بیابید که توسط مشتریانی خریداری شده‌اند که مجموع هزینه آن‌ها بالاتر از میانگین است. برای هر آهنگ، شناسه، نام، تعداد مشتریان یکتا که آن را خریداری کرده‌اند و مجموع درآمد حاصل از آن خریدها را نمایش دهید. تنها آهنگ‌هایی را در نظر بگیرید که حداقل توسط یک مشتری واجد این شرایط خریداری شده‌اند و نتایج را بر اساس تعداد مشتریان یکتا به ترتیب نزولی مرتب کنید.

TrackId	Name	num_customers	total_revenue
---------	------	---------------	---------------

۳- تعیین کنید که کدام کارمندان مشتریانی را مدیریت می‌کنند که میانگین هزینه‌های آن‌ها بالاتر از میانگین جهانی هزینه مشتریان است. برای هر یک از این کارمندان، شناسه، نام، نام خانوادگی، تعداد مشتریانی که مدیریت می‌کنند و میانگین هزینه مشتریان آن‌ها را فهرست کنید. نتایج را بر اساس میانگین هزینه مشتریان به ترتیب نزولی مرتب نمایید.

EmployeeId	FirstName	LastName	num_customers	avg_customer_spending
------------	-----------	----------	---------------	-----------------------

۴- پلی‌لیست‌هایی را شناسایی کنید که شامل آهنگ‌هایی از حداقل سه ژانر مختلف و حداقل سه هنرمند متفاوت هستند. برای هر پلی‌لیست واجد شرایط، شناسه، نام و یک رشته متشکل از ترکیب‌های یکتای «ژانر توسط هنرمند» (مثلاً "راک توسط بیتلز؛ پاپ توسط مدونا") را فهرست کنید. نتایج را بر اساس تعداد ژانرهای متمایز به ترتیب نزولی مرتب نمایید.

playlistId	Name	genre_artist_combinations
------------	------	---------------------------

۵- کشورهای را بیابید که میانگین میانگین فاکتور هر مشتری در آن‌ها بالاتر از میانگین جهانی بوده و حداقل دو مشتری در آن وجود دارد. برای هر یک از این کشورها، نام کشور، تعداد مشتریان و میانگین

میانگین فاکتورهای مشتریان را فهرست کنید. نتایج را بر اساس این میانگین به ترتیب نزولی مرتب نمایید.

Country	num_customers	avg_avg_invoicetotal
---------	---------------	----------------------

۶- شناسه کارمندی که مدیر هستند (دستکم یک نفر به آنها گزارش میدهد) به همراه نام کامل آن مدیر و تعداد نفراتی که مستقیم یا غیرمستقیم به او گزارش میدهند را برگردانید. نتایج را براساس تعداد افرادی که هر فرد مدیریت میکند به صورت نزولی مرتب کنید.

EmployeeId	ManagerName	total_managed
------------	-------------	---------------

## FIFA World Cup

کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های `part1.sql` و `part2.sql` و ... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در `root` فایل zip قرار بگیرند.

شما می‌توانید از این فایل **Dump دیتابیس** برای تست کوئری‌های خود استفاده کنید.

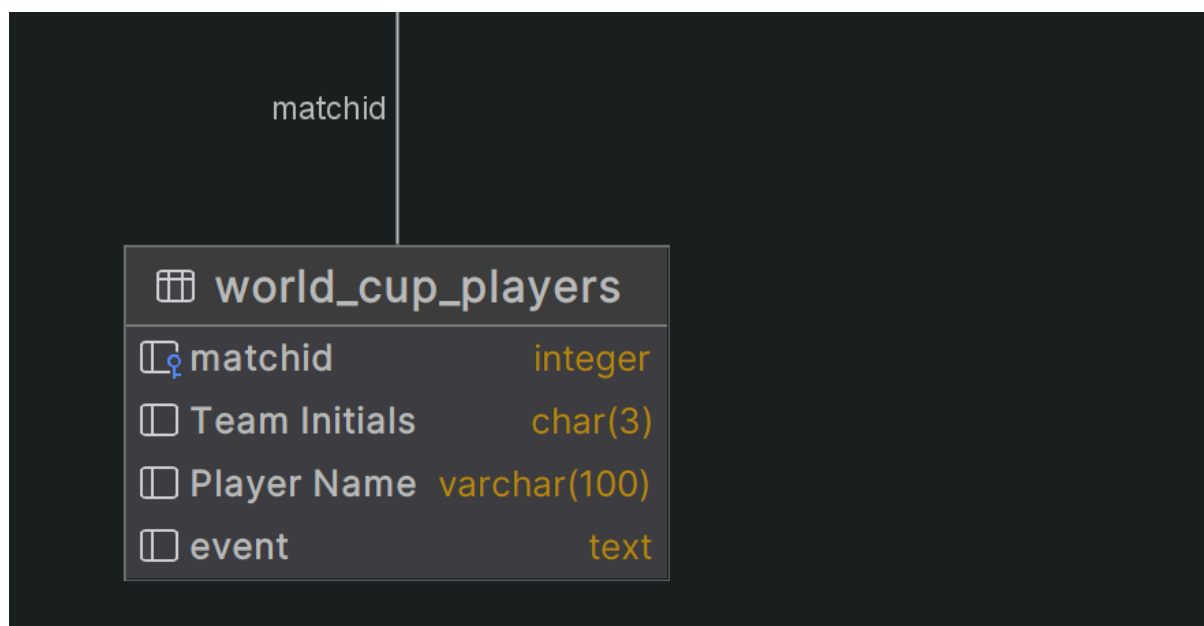
می‌توانید از این دستور استفاده کنید تا فایل Dump را وارد دیتابیس خود کنید:

```
psql -U <Username> -d <Database_name> -f FIFA.dump
```

ساختار جداول این دیتابیس به این صورت است:

WorldCups	
Year	integer
Country	text
Winner	text
Runners-Up	text
Third	text
Fourth	text
GoalsScored	integer
QualifiedTeams	integer
MatchesPlayed	integer
Attendance	bigint

world_cup_matches	
year	smallint
stage	varchar(50)
Home_Team_Name	varchar(100)
Home_Team_Goals	integer
Away_Team_Goals	integer
Away_Team_Name	varchar(100)
win_conditions	varchar(100)
attendance	integer
Half-time Home Goals	integer
Half-time Away Goals	integer
referee	varchar(100)
Home_Team_Initials	char(3)
Away_Team_Initials	char(3)
matchid	integer



در ستون event مقادیر وارد شده با این منظور است:

G ==> Goal  
 OG ==> Own Goal  
 Y ==> Yellow Card  
 R ==> Red Card  
 RSY ==> Red Card by second yellow  
 P ==> Penalty  
 MP ==> Missed Penalty  
 I ==> Substitution In  
 O ==> Substitute Out

به کوئری‌های زیر با توجه به ساختارهای داده شده جواب دهید:

۱- مسابقاتی که در آن‌ها بیش‌تر از ۵ گل زده شده است.

stage	Home_Team_Name	Home_Team_Goals	Away_Team_Goals	Away_Team_Name
-------	----------------	-----------------	-----------------	----------------

۲- مسابقاتی که بین سال‌های ۱۹۳۸ تا ۲۰۰۶ به وقت اضافه رفته‌اند.

matchid	year	stage	Home_Team_Name	Away_Team_Name
---------	------	-------	----------------	----------------

۳- داورانی که بیشترین کارت قرمز را داده‌اند.

referee	totalcards
---------	------------

۴- میانگین تماشاگر بازی‌های کشور میزبان به تفکیک تیم و سال که به صورت نزولی مرتب شده است.

Country	Year	avgparticipantspermatch
---------	------	-------------------------

۵- تعداد بازی‌های هر سال که در نیمه اول یک تیم جلو بوده اما در پایان بازی تیم مقابل برنده شده است.  
(از ضربات پنالتی که بعد از دقیقه 120 است صرف نظر شود)

year	count
------	-------

۶- بازیکنانی که در مسابقات میزبانی کشور خود (کشور خود باید میزبان آن دوره از مسابقات باشد)، کارت گرفته‌اند.

Player Name	Year	Team Initials	Country	event
-------------	------	---------------	---------	-------

## IMDB

کدهای مربوط به هر بخش را در فایل‌هایی با نام‌های part1.sql و part2.sql و ... قرار دهید و همه را با هم zip کنید و آپلود کنید. دقت کنید که پاسخ‌ها داخل فولدر دیگری نباشند و در root فایل zip قرار بگیرند.

بخشی از داده‌های فیلم‌ها، مشابه با سایت IMDB به صورت جدول برای شما قرار داده شده است. پایگاه داده این بخش به شکل زیر طراحی شده است:

```
1 | users(user_id, first_name, last_name, Email)
2 |
3 | actors(actor_id, name)
4 |
5 | movies(movie_id, title, director, budget, summary, countary)
6 |
7 | reviews(review_id, user_id, movie_id, review_text, score)
8 |
9 | act(actor_id, movie_id)
```

شما می‌توانید از این دیتاها برای تست کوئری‌های خود استفاده کنید.

### توضیحات مختصر :

در موجودیت user کاربرهای سایت نگهداری شده‌اند و ویژگی user\_id برای این موجودیت Primary key است.

در موجودیت actors اسامی بازیگران سینما نگهداری شده‌اند و ویژگی actor\_id برای این موجودیت Primary key است.

در موجودیت movies برای هر فیلم، اسم آن (title)، کارگردان، بودجه‌ی ساخت، خلاصه‌ای از داستان فیلم و کشور مبدا آن فیلم نگهداری شده‌اند. movie\_id برای این موجودیت Primary key است.

در موجودیت reviews، متن‌هایی که کاربران برای فیلم نوشته‌اند، ذخیره شده است. در کنار بعضی از نقدها، امتیازی وجود دارد که کاربر در نقد review\_id به فیلم movie\_id داده است. review\_id برای این موجودیت Primary key است و user\_id و movie\_id برای این موجودیت، Foreign key هستند.

در موجودیت `act`، برای بازیگران مختلف، فیلم‌هایی که در آن بازی کرده‌اند را ذخیره کرده‌ایم. `actor_id` و `movie_id` برای این موجودیت `Foreign key` اند و زوج‌مرتبشان `Primary key` است.

به کوئری‌های زیر با توجه به ساختارهای داده شده جواب دهید:

۱- فیلم‌هایی که در `title` آن‌ها عبارت `Simin` وجود دارد.

۲- فیلم‌هایی که در `summary` یا `title` آن‌ها حداقل دو بار از عبارات `Gotham` یا `Batman` استفاده شده باشد.

۳- تابعی بنویسید که `actor_id` را به عنوان ورودی بگیرد و در خروجی تعداد کشورهای مختلفی که این بازیگر در آن فیلم بازی کرده است را به عنوان خروجی بدهد.

۴- با استفاده از تابعی که در قسمت قبل ساختید، بازیگران را به ترتیب نزولی تعداد کشورهای مختلفی که در آن فیلم بازی کرده‌اند خروجی دهید.

۵- جدولی خروجی دهید که برای هر کارگردان میانگین بودجه‌ی فیلم‌هایی که ساخته است در آن نوشته شده باشد. (به ترتیب این میانگین به صورت نزولی خروجی دهید)

۶- آقای `hamid farzane` که از کاربران سایت `IMDB` است، ایمیل خود را در سایت به `farzHMM@gmail.com` تغییر داده است. این تغییر را در پایگاه داده اعمال کنید. صحت تغییر اعمال شده را با یک دستور `select` بررسی کنید.