

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر



پایگاه های داده

فاز دوم گروه ۱۹

امیر حسین باقری ۹۸۱۰۵۶۲۱
بردیا کلانتری ۹۹۱۷۰۴۵۶
امیر محمد ایمانی ۹۸۱۰۹۶۸۹
استاد:

دکتر امینی

فهرست مطالب

۲	بخش اول	۱
۲	۱.۱ ساختن جداول	
۲	۲.۱ تولید دیتا	
۲	بخش دوم	۲
۳	۱.۲ کوئری اول	
۴	۲.۲ کوئری دوم	
۴	۳.۲ کوئری سوم	
۵	۴.۲ کوئری ۴ام	
۸	۵.۲ کوئری پنجم	
۸	۶.۲ کوئری ششم	
۹	۷.۲ کوئری هفتم	
۹	بخش سوم	۳
۹	۱.۳ view	
۱۰	۲.۳ index	
۱۱	۳.۳ trigger	

۱ بخش اول

۱.۱ ساختن جداول

کدهای ساختن جدول ها در فایل

create_tables.sql

قرار دارند. تنها تغییر داده شده در جدول نسبت به فاز یک اضافه کردن آی دی به مدیر بود این کار برای راحت تر شدن کار نسبت به کار کردن با یوزرنیم است.

a_descriptive answerid txt questionid	supervisor id username password	options questionid number txt	survey id managerid flightid active creation_date start_date end_date airlineid	ticket_type typeid type_string price	approving supervisorid questionid approve_date	q_multiple_choice questionid txt a_multi_choice answerid questionid number		
airline id name managerID	q_descriptive questionid txt	flight flight_number airlineid	ticket ticket_number passenger_id flight_number passport_number seat_number ticket_date ticket_typeid surveyid participated	question id mandatory ticket_typeid surveyid	answer id ticketid questionid creation_date	manager id username password	passenger id firstname lastname	assistance adminid assistanceid surveyid

شکل ۱: جداول تغییر داده شده و نهایی در عکس بالا هستند.

۲.۱ تولید دیتا

کدهای تولید دیتا در

create_data.sql

قرار دارند.

۲ بخش دوم

کدهای هر ۷ بخش بخش دوم در فایل هایی به نام های *procedure num* قرار دارند که عدد به عنوان پسوند آمده است. دقت کنید که ورژن *mysql* تست شده برای این کدها

mysql Ver 8.0.31-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))

است.

۱.۲ کووری اول

به دلیل نوع ذخیره داده ها در جداول برای سوالات چند گزینه ای و سوالات تشریحی دو تابع نوشته ایم. تابع

CALL participate_surveyqd

برای سوالات تشریحی و تابع

CALL participate_surveyqm

برای سوالات تستی نتیجه اجرای ۲ تابع در زیر آمده است.

CALL participate_surveyqd(12344,4);

ID int	Question varchar
1	how do you describe fligh assistances attitude?
2	how was the meal?
3	how pleasent you found air condotioner?
4	did you had a good seat?
5	how did you found noise in airplain?
6	how do you describe your financial state?
7	How often do you fly?

شکل ۲: سوالات تشریحی

CALL participate_surveyqm(12344,4);

ID int	Question varchar	num int	choice varchar
1	Departure and / or arrival time was more convenient?	1	Very important
2	Departure and / or arrival time was more convenient?	2	Important
3	Departure and / or arrival time was more convenient?	3	Neutral
4	Departure and / or arrival time was more convenient?	4	Unimportant
5	Departure and / or arrival time was more convenient?	5	Not at all important
6	Frequent flyer program?	1	Very important
7	Frequent flyer program?	2	Important
8	Frequent flyer program?	3	Neutral
9	Frequent flyer program?	4	Unimportant
10	Frequent flyer program?	5	Not at all important
11	Inflight services are better?	1	Very important
12	Inflight services are better?	2	Important
13	Inflight services are better?	3	Neutral
14	Inflight services are better?	4	Unimportant
15	Inflight services are better?	5	Not at all important
16	Travel agent / company travel department recommendation?	1	Very important
17	Travel agent / company travel department recommendation?	2	Important
18	Travel agent / company travel department recommendation?	3	Neutral
19	Travel agent / company travel department recommendation?	4	Unimportant
20	Travel agent / company travel department recommendation?	5	Not at all important

شکل ۳: سوالات چند گزینه ای

۲.۲ کوئری دوم

در اینجا تابع

CALL optionals_question

نوشته ایم که نتیجه آن را در عکس بدی مشاهده می کنید.

CALL optionals_question(1,"2020-01-23 12:45:56", "2021-12-23 12:45:56");

	QID int	Qtxt varchar	opt int	txt varchar	COUNT(*) bigint
1	16	Departure and / or arrival time	1	Very important	1
2	17	Frequent flyer program?	2	Important	1
3	18	Inflight services are better?	3	Neutral	1
4	19	Travel agent / company travel	1	Very important	2
5	16	Departure and / or arrival time	2	Important	1
6	17	Frequent flyer program?	3	Neutral	1
7	18	Inflight services are better?	4	Unimportant	1

شکل ۴: تعداد پاسخ به سوالات چند گزینه ای

۳.۲ کوئری سوم

در این جا تابع

dec_question

نوشته ایم که نتایج آنرا در دو عکس بعدی برای دو کلمه متفاوت مشاهده می کنید.

CALL dec_question(1,"if");

	customer_ticketID int	Q varchar	Ans varchar
1	1	any comments?	If someone is going to ask you the same old everyday

شکل ۵: با کلمه if

CALL dec_question(1,"better");

	customer_ticketID int	Q varchar	Ans varchar
1	1	how do you describe your financial state?	fine but could be better

شکل ۶: کلمه با better

۴.۲ کوئری ۴م

برای این بخش تابع

qmmanageraccesss

نوشته ایم که نمونه خروجی آن را در شکل زیر مشاهده می کنید. دقت کنید که برای این بخش مدیر ۱ و ۳ را دستیار کرده ایم بنابراین برای هر دو مدیر تست می کنیم و نتایج یکسان را مشاهده می کنید. دقت کنید که مدیر یک *admin* است.

```
CALL qmmanageraccesss(1);
```

</

CALL qmmanageraccesss(3);				
<div> <div> <div>+</div> <div>🔒</div> <div>🔍</div> <div>Input to filter result</div> </div> <div> <div>⚙️</div> <div>Free</div> <div>📧</div> <div>🔄</div> <div>+</div> <div>+</div> <div>🗑️</div> <div>🔍</div> <div>1</div> </div> </div>				
<input checked="" type="checkbox"/>	🔍	id int	type_string varchar	COUNT(*)/@COUNT newdecimal
	1	1	business	0.3333
	2	2	business	0.3333
	3	3	business	0.3333
	4	4	business	0.3333
	5	5	business	0.3333
	6	6	business	0.3333
	7	7	business	0.3333
	8	8	business	0.3333
	9	9	business	0.3333
	10	10	business	0.3333
	11	11	business	0.3333
	12	16	business	0.3333
	13	17	business	0.3333
	14	18	business	0.3333
	15	19	business	0.3333
	16	12	economy	0.1667
	17	13	economy	0.1667
	18	14	economy	0.1667
	19	15	economy	0.1667
	20	20	economy	0.1667
	21	21	economy	0.1667

۵.۲ کوئری پنجم

برای این بخش دو تابع برای سوالات تشریحی و تستی جدا نوشته ایم تابع

not_approvedqd

برای سوالات تشریحی

not_approvedqm

برای سوالات تستی نتایج هر کدام از این کوئری ها در دو عکس بعدی آمده است.

```
CALL not_approvedqd();
```

questionid int	txt varchar
1	4
	how was pilot infomration?

شکل ۹: تشریحی

```
CALL not_approvedqm();
```

questionid int	txt varchar	questionid int	number int	txt varchar
1	21	21	1	Very importar
2	21	21	2	Important
3	21	21	3	Neutral
4	21	21	4	Unimportant
5	21	21	5	Not at all imp
6	22	22	1	Didn't Use
7	22	22	2	No Wait
8	22	22	3	0-5 min
9	22	22	4	11-20 min
10	22	22	5	20+ min

شکل ۱۰: چند گزینه ای

۶.۲ کوئری ششم

برای این بخش تابع *stats* را نوشته ایم که خروجی آن را در تصویر بعدی مشاهده می کنید.

`CALL stats();`

		AirlineName varchar	participant_number bigint
1		mahan	2
2		qatar	1

شکل ۱۱: آمار شرکت ها

۷.۲ کوئری هفتم

برای این بخش تابع

`statsquestionqd`

نوشته شده است. تنها نکته این تابع اضافه کردن

`ORDER BY RAND();`

به انتها کوئری است. در بین دیتا ها ۳ نفر در سوال یک شرکت کرده اند که یکی از آنها ۲ پرواز یکی ۷ پرواز و یکی دیگر تنها یک پرواز داشته است. بنابراین نتیجه خروجی به شرح زیر است.

`CALL statsquestionqd(1);`

		passenger_id int	txt varchar
1		1	it was ok ali

۳ بخش سوم

۱.۳ view

ابتدا view را مطابق زیر تعریف می کنیم.

```

create view view1 as
    SELECT survey.id, survey.start_data, survey.end_date
    , survey.createtion_date, CASE survey.active
        WHEN 1 THEN 'YES'
        ELSE 'NO'
    end ACTIVE, COUNT(DISTINCT question.id) as ALLQ,
    COUNT(DISTINCT ticket.ticket_number)/(SELECT COUNT(*)
    FROM ticket as t
    WHERE t.flight_number = survey.flightid) AS participantsRatio

FROM ticket INNER JOIN survey on ticket.surveyid = survey.id
INNER JOIN question on question.surveyid = survey.id
GROUP BY survey.id, survey.start_data,
survey.end_date,
survey.createtion_date,
survey.active;

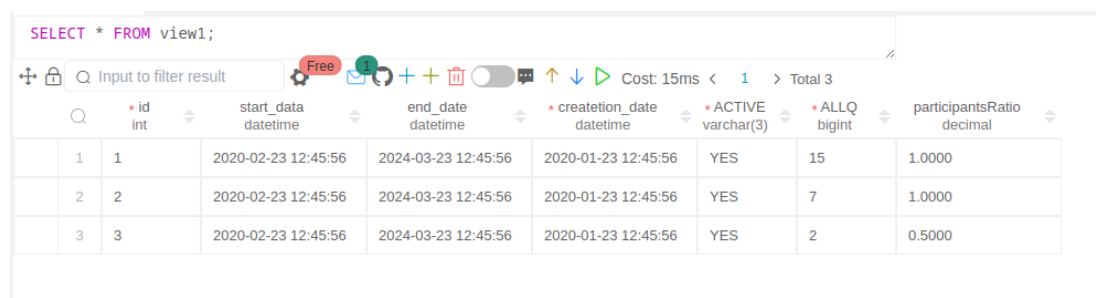
SELECT * FROM view1;

```

نتیجه خروجی به شرح زیر است. کد نیز در فایل

view.sql

قرار دارد.



	id int	start_data datetime	end_date datetime	createtion_date datetime	ACTIVE varchar(3)	ALLQ bigint	participantsRatio decimal
1	1	2020-02-23 12:45:56	2024-03-23 12:45:56	2020-01-23 12:45:56	YES	15	1.0000
2	2	2020-02-23 12:45:56	2024-03-23 12:45:56	2020-01-23 12:45:56	YES	7	1.0000
3	3	2020-02-23 12:45:56	2024-03-23 12:45:56	2020-01-23 12:45:56	YES	2	0.5000

۲.۳ index

در این بخش تنها به بیان علت می پردازیم که چرا چنین شاخص هایی تعریف کرده ایم. برای *creation date* از *B tree* استفاده می کنیم برای بقیه ستون های استفاده شده چون حالت تساوی را بررسی می کنیم از *HASH* استفاده می کنیم. کد های این بخش نیز در

index.sql

قرار دارد.

```

CREATE INDEX ansdate ON answer (createtion_date)
USING BTREE;

```

```

CREATE INDEX a_multi_choice1 ON a_multi_choice (questionid)
USING HASH;

```

```

CREATE INDEX a_multi_choice2 ON a_multi_choice (answerid)
USING HASH;
CREATE INDEX a_multi_choice3 ON a_multi_choice (number)
USING HASH;

CREATE INDEX options1 ON options (questionid)
USING HASH;
CREATE INDEX options2 ON options (number)
USING HASH;

CREATE INDEX survey1 ON survey (managerid)
USING HASH;

CREATE INDEX answer1 ON answer (ticketid)
USING HASH;

CREATE INDEX q_multiple_choice1 ON q_multiple_choice (questionid)
USING HASH;

```

trigger ۳.۳

این بخش پیچیدگی خاصی ندارد و تعریف کردن *trigger* برای این امر بسیار ساده است. دو رهانا تعریف می کنیم که یکی هنگام درج و دیگری هنگام آپدیت اجرا می شوند. برای حالت آپدیت چک می کنیم که تاریخ پایان جدید از شروع عقب تر نباشد همچنین تاریخ شروع جدید از تاریخ پایان قبلی جلو تر باشد. همچنین تاریخ از قبل نگذشته باشد و همچنین با تاریخ های فعلی تداخل نداشته باشد. برای حالت درج نیز مانند بالا تنها دیگر نیاز به چک کردن با حالت قدیمی نداریم. کدهای این بخش نیز داخل

trigger.sql

قرار دارند.

```

USE dbproject;
DROP trigger IF EXISTS trgupdate;
create trigger trgupdate
before update
on survey
for each row
IF (NEW.start_data > NEW.end_date) OR
(NEW.start_data > OLD.end_date) OR
(NEW.start_data < some(select end_date
from survey where survey.id <> OLD.id)) THEN
SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = 'Error!';
END IF;

```

```
DROP trigger IF EXISTS trgininsert;
create trigger trgininsert
before insert
on survey
for each row
IF (NEW.start_data > NEW.end_date) or
(NEW.start_data < some(select end_date from survey)) THEN
SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT = 'Error';
END IF;
```