

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر - گروه ۶

پروپوزال برای پروژه‌ی درس پایگاه داده
رشته: مهندسی کامپیوتر

عنوان:

بوت کمپ الگوریتم کوئرا

استاد درس:

مهدی دادبخش

نگارش:

دنیا نوابی ۹۹۱۷۰۵۹۷

شایان صالحی ۹۹۱۰۵۵۶۱

ایمان محمدی ۹۹۱۰۲۲۰۷

اسفند ماه ۱۴۰۱

ترم بهار ۱۴۰۲

چکیده

در این پروژه قصد پیاده‌سازی پایگاه داده‌ی بوت‌کمپ کوئرا کالج را داریم. در این داکيومت نیز، به شرح سیستم، نیازها و پیاده‌سازی سیستم می‌پردازیم.

فهرست مطالب

صفحه	عنوان
۲	فصل ۱ مقدمه
۳	۱-۱. مقدمه.....
۴	فصل ۲ موجودیت‌ها
۴	۲-۱. تعریف موجودیت‌ها.....
۵	فصل ۳ شرح کلی و رابطه‌ها
۶	۳-۱. روابط سیستم.....
۸	فصل ۴ شرح صفات هر موجودیت
۹	۴-۱. صفات موجودیت‌های سیستم.....
۱۵	فصل ۵ جداول و درج داده‌ها
۱۶	۵-۱. جداول و درج داده‌ها.....
۳۲	فصل ۶ لیست عملیات و گزارشات پروژه
۳۳	۶-۱. عملیات و گزارشات پروژه.....
۵۱	بخش امتیازی
۵۲	۶-۱. اتصال به زبان برنامه‌نویسی پایتون.....
۶۶	مراجع

فصل ۱

مقدمه

۱-۱. مقدمه

در ابتدا می‌خواهیم یک معرفی کوتاهی از بوت کمپ داشته باشیم. بوت کمپ یک دوره‌ی کوتاه مدت با هدف کسب مهارت برای افراد در یک حوزه‌ی خاص است. در این فضا افراد شرکت‌کننده توسط یک سری افراد متخصص در آن حوزه آموزش می‌بینند و با انجام تمرین‌ها و پروژه‌های از پیش تعیین شده، مهارت فنی و تیمی‌شان را افزایش می‌دهند. همچنین ممکن است در طول دوره یا انتهای آن، یک سری ارزیابی از مهارت‌های یاد گرفته شده صورت گیرد و ممکن است براساس نتایج این آزمون‌ها، یک مدرک خاص به آنها داده شود و یا به شرکت‌های صاحب قرارداد با بوت کمپ معرفی شوند. به همین دلیل بوت کمپ این فضا را برای دانشجویان یا فارغ‌التحصیلان مهیا می‌کند، که بتوانند در فیلد مورد علاقه‌شان، شانس ورود به بازار کار را افزایش دهند.

حال کوئرا کالج یک تعدادی بوت کمپ برای افراد علاقه‌مند در زمینه‌های مختلف کامپیوتر مهیا کرده است. هدف این پروژه، بررسی پایگاه داده بوت کمپ الگوریتم کوئرا می‌باشد. این بوت کمپ در یک دوره چهار ماهه برگزار می‌شود. در سه ماه اول، این دوره به صورت موازی به شرکت‌کنندگان، آموزش و تعدادی تمرین برای تحویل داده می‌شود و در ماه آخر نیز شرکت‌کنندگان در حال انجام پروژه (در دو فاز) و آمادگی برای آزمون نهایی خواهند بود.

فصل ۲

موجودیت‌ها

۲-۱. تعریف موجودیت‌ها

در این بوت‌کمپ ما یک سری موجودیت داریم. موجودیت‌ها به این صورت خواهند بود: رئیس بوت‌کمپ، منتورها، اساتید، افراد ثبت‌نامی، شرکت‌کنندگان، قبول شدگان، تمرین‌ها، پروژه، آزمون‌ها، شرکت‌های صاحب قرارداد، کلاس، سابمیت‌های نهایی، سابمیت‌های هر شرکت‌کننده.

فصل ۳

شرح کلی و رابطه‌ها

۱-۳. روابط سیستم

در هر دوره‌ی بوتکمپ یک رئیس داریم که مسئولیت وی، هماهنگی با اساتید و شرکت‌های صاحب قرارداد است. همچنین ایشان وظیفه نظارت بر عملکرد منتورها را نیز برعهده دارد. تعدادی استاد نیز برای دوره داریم که وظیفه‌ی آن‌ها، آموزش بخش‌های مختلف مربوط به دوره به شرکت‌کنندگان است.

شرکت‌های صاحب قرارداد در انتهای دوره با شرکت‌کنندگان منتخب، مصاحبه می‌کنند که این منتخبین توسط رئیس دوره با توجه به نتیجه عملکردشان معرفی می‌شوند تا در صورت قبولی در مصاحبه و تایید شرکت‌کننده، استخدام شوند.

منتورها وظیفه‌ی طراحی تمرین‌ها، پروژه‌ها و آزمون‌هایی را بر عهده دارند. همچنین هر منتور، مسئولیت راهنمایی یک گروه شرکت‌کننده شامل چهار یا پنج نفر (بسته به تعداد کل شرکت‌کنندگان) در طول دوره برعهده دارد.

افراد ثبت‌نامی، بر اساس شرایط از پیش تعیین شده برای شرکت در دوره نظیر: سن، مدرک تحصیلی و غیره به دوره وارد می‌شوند و اگر تعداد آنها بیشتر از بیست نفر بود، بیست نفری که زودتر از مابقی افراد ثبت‌نام کردند، مجاز به شرکت در بوتکمپ خواهند شد، همچنین افرادی با مدرک بالاتر از کارشناسی ارشد پذیرفته نخواهند شد.

هر شرکت‌کننده برای این‌که جزو منتخبین دوره باشد، می‌بایست تمرین‌ها و پروژه‌های طرح شده را تحویل دهد و در آزمون‌هایی، عملکرد قابل قبولی نسبت به مابقی شرکت‌کنندگان داشته باشد.

برای هر تمرین دو جدول سابمیت خواهیم داشت، یک جدول برای هر فرد زیرا هر فرد می‌تواند برای هر تمرین چندین سابمیت داشته باشد و یک جدول سابمیت‌های نهایی خواهیم داشت که بالاترین نمره از سابمیت‌های هر فرد در آن جدول در زمان تعیین شده برای آن تمرین قرار داده شده و برای کل شرکت‌کنندگان خواهد بود. همچنین نمرات نهایی تمرین‌ها

براساس جدول سابعیت‌های نهایی تعیین می‌شود. تمارین هر هفته ریلیز می‌شوند و به مدت یک هفته تا ریلیز تمرین بعدی مهلت ارسال خواهند داشت.

پروژه نیز بسیار شبیه تمرین‌ها خواهد بود با این تفاوت که دو فاز برای شرکت‌کنندگان تعریف می‌شود و برای هرفاز مدت زمان ده روز برای تحویل آن در نظر گرفته می‌شود.

یک آزمون نهایی خواهیم داشت که تاریخ آن از ابتدای دوره مشخص بوده و یک فرجه‌ی ده روزه پس از اتمام پروژه برای آمادگی شرکت‌کنندگان در آزمون خواهد داشت

فصل ۴

شرح صفات هر موجودیت

۲-۳. صفات موجودیت‌های سیستم

رئیس بوت کمپ:

نام و نام خانوادگی، مدرک تحصیلی، شماره تلفن همراه، شناسه رئیس (کلید اصلی)
اساتید:

نام و نام خانودگی، مدرک تحصیلی، شماره تلفن همراه، ایمیل، کد شناسه استاد (کلید اصلی)
منتورها:

نام و نام خانودگی، مدرک تحصیلی، شماره تلفن همراه، کد شناسه منتور (کلید اصلی)،
شناسه پروژه (کلید خارجی)، شناسه تمرین (کلید خارجی)، شناسه آزمون (کلید خارجی)،
شناسه رئیس (کلید خارجی)
افراد ثبت‌نامی:

نام و نام خانودگی، مدرک تحصیلی، شماره تلفن همراه، زمان ثبت‌نام، کد ملی (کلید اصلی)
شرکت‌کنندگان:

نام و نام خانودگی، مدرک تحصیلی، شماره تلفن همراه، نمره آزمون، کد ملی (کلید اصلی و خارجی)، شناسه آزمون (کلید خارجی)
قبول شدگان:

نام و نام خانودگی، مدرک تحصیلی، شماره تلفن همراه، شناسه قبولی (کلید اصلی و خارجی)
شرکت‌های صاحب قرارداد:

نام شرکت، شماره تلفن ثابت، حقوق پیشنهادی هر شرکت، نوع کار (حضور یا دورکاری)،

زمینه کار، کد شرکت (کلید اصلی)، شناسه رئیس (کلید خارجی)

تمرین‌ها:

شماره‌ی تمرین، زمان انتشار تمرین، مهلت ارسال پاسخ‌ها، موضوع تمرین، شناسه تمرین

(کلید اصلی)

پروژه‌ها:

شماره‌ی فاز پروژه، زمان انتشار فاز پروژه، مهلت ارسال فاز پروژه، موضوع پروژه، شناسه

پروژه (کلید اصلی)

آزمون نهایی:

تعداد سوالات، مدت زمان آزمون، تاریخ برگزاری آزمون، شناسه آزمون (کلید اصلی)

ارسال‌های اولیه:

نمره‌ی ارسال، تاریخ ارسال - ساعت ارسال (کلید اصلی)، کد ملی شرکت‌کننده (کلید

خارجی)

ارسال نهایی:

نمره‌ی ارسال، تاریخ ارسال - ساعت ارسال (کلید اصلی)، کد ملی شرکت‌کننده (کلید

خارجی)

$(\prime \prime)$

فصل ۵

جداول و درج داده‌ها

در این قسمت به قسمت مربوط به کد ساخت جداول درج داده‌ها در آن می‌پردازیم.

ساخت دیتابیس اصلی و تغییر آن برای زبان فارسی:

```
-- ایجاد پایگاه داده بوتکمپ الگوریتم کوئرا
Create Database queraAlgorithmBootcamp
-- on( Name='bootcamp_data',FileName='G:\database\quera_algorithm_bootcamp\bootcamp_data.mdf',Size=5 )
-- log on( Name='bootcamp_log',FileName='G:\database\quera_algorithm_bootcamp\bootcamp_log.ldf' )
Go

-- تنظیم تلفیق حروف برای زبان فارسی
Alter Database queraAlgorithmBootcamp Collate Persian_100_CI_AI
Go

USE queraAlgorithmBootcamp
Go
```

ساخت جدول آزمون نهایی و درج داده در آن:

```
17 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
18 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Final_exam')
19 CREATE TABLE Final_exam
20 {
21     Exam_id nvarchar(50) primary key, -- شناسه آزمون نهایی
22     Exam_date date not null, -- تاریخ برگزاری آزمون
23     Nom_of_Exam tinyint not null, -- تعداد سوالات
24     Exam_time time not null, -- مدت زمان آزمون
25 }
26 Go
27
28 -- درج داده در جدول آزمون نهایی
29 Insert Into Final_exam
30 Values
31 (N'آزمون نهایی', '2023-07-01', 7, '12:00:00')
32 Go
```


ساخت جدول افراد ثبت‌نامی و درج داده در آن:

```

34 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز --
35 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Registered.people')
36 CREATE TABLE Registered.people -- جدول افراد ثبت نامی
37 (
38     National_id nchar(10) PRIMARY KEY, -- کد ملی افراد ثبت نامی
39     Mobile_number char(11) NOT NULL, -- شماره تلفن همراه افراد ثبت نامی
40     First_name nvarchar(30) NOT NULL, -- نام افراد ثبت نامی
41     Last_name nvarchar(30) NOT NULL, -- نام خانوادگی افراد ثبت نامی
42     Degree nvarchar(20) NULL, -- مدرک افراد ثبت نامی
43     Register_time datetime NOT NULL, -- زمان ثبت نام افراد ثبت نامی
44     Exam_id nvarchar(50) NOT NULL,
45     Exam_score int not null,
46 )
47 GO

48 INSERT INTO Registered.people
49 -- VALUES
50 ('0352465468', '09120312485', N'محمد', N'احمد', N'کارشناس ارشد', '2023-01-02 16:20:11', N'آزمون نهایی', 100),
51 ('0784668419', '09174412481', N'رضا', N'مصطفی', N'کارشناس', '2023-01-02 18:20:22', N'آزمون نهایی', 88),
52 ('0318966291', '09874398465', N'یاسین', N'رضاییان', N'کارشناسی ارشد', '2023-01-02 18:10:12', N'آزمون نهایی', 66),
53 ('0372865461', '09345398785', N'دنیا', N'احمدی', N'کارشناسی', '2023-01-03 19:10:10', N'آزمون نهایی', 25),
54 ('0973274052', '09135394143', N'ایمان', N'پوری', N'دکتری', '2023-01-03 22:20:15', N'آزمون نهایی', 90),
55 ('0973274055', '09145394231', N'نجد', N'رضایی', N'دکتری', '2023-01-04 12:20:15', N'آزمون نهایی', 27),
56 ('0973274050', '09178924232', N'سارا', N'سلیمی', N'دکتری', '2023-01-02 15:50:13', N'آزمون نهایی', 90),
57 ('1294274044', '09164744242', N'فرمان', N'شریفی', N'کارشناسی', '2023-01-04 12:50:53', N'آزمون نهایی', 85),
58 ('1852465470', '09120312586', N'محمود', N'میرزایی', N'کارشناسی', '2023-01-02 08:20:11', N'آزمون نهایی', 68),
59 ('2084668420', '09174412582', N'فرهاد', N'صدیقی', N'کارشناسی', '2023-01-02 11:20:22', N'آزمون نهایی', 50),
60 ('3058966292', '09874398566', N'زهره', N'کریمی', N'کارشناسی ارشد', '2023-01-02 14:10:12', N'آزمون نهایی', 40),
61 ('4062865461', '09345398885', N'شیدا', N'شریفی', N'کارشناسی', '2023-01-02 15:10:10', N'آزمون نهایی', 35),
62 ('5063274052', '09135394243', N'علی', N'محمدی', N'دکتری', '2023-01-02 16:20:15', N'آزمون نهایی', 75),
63 ('6063274055', '09145394331', N'بهنام', N'رضایی', N'دکتری', '2023-01-03 09:20:15', N'آزمون نهایی', 25),
64 ('7063274050', '09178924332', N'لیلا', N'خوردی', N'فیلیم', '2023-01-03 12:50:13', N'آزمون نهایی', 30),
65 ('8074274044', '09164744342', N'علیر', N'رستمی', N'کارشناسی', '2023-01-03 15:50:53', N'آزمون نهایی', 90),
66 ('9004274047', '09164744345', N'فاطمه', N'علیراده', N'کارشناسی', '2023-01-03 18:20:23', N'آزمون نهایی', 100),
67 ('1093274058', '09178924339', N'پریسا', N'محسنی', N'دکتری', '2023-01-04 09:30:25', N'آزمون نهایی', 90),
68 ('2103274060', '09178924341', N'حارث', N'رضایی', N'دکتری', '2023-01-04 12:20:35', N'آزمون نهایی', 40),
69 ('3103274061', '09178924342', N'آیدا', N'شیرازی', N'کارشناسی', '2023-01-04 15:20:45', N'آزمون نهایی', 88),
70 ('4103274062', '09178924343', N'حمید', N'زینتی', N'کارشناسی ارشد', '2023-01-04 17:20:55', N'آزمون نهایی', 98),
71 ('5103274063', '09178924344', N'مینا', N'میرزاده', N'کارشناسی ارشد', '2023-01-05 09:30:05', N'آزمون نهایی', 100),
72 ('6103274064', '09178924345', N'پیمان', N'پورانی', N'کارشناسی', '2023-01-05 12:30:15', N'آزمون نهایی', 99),
73 ('7103274065', '09178924346', N'روینده', N'محمدی', N'کارشناسی', '2023-01-05 15:30:25', N'آزمون نهایی', 65),
74 ('8103274066', '09178924347', N'کایو', N'کریمی', N'کارشناسی', '2023-01-05 18:30:35', N'آزمون نهایی', 69),
75 ('9103274067', '09178924348', N'آرش', N'امشاری', N'کارشناسی ارشد', '2023-01-05 21:30:45', N'آزمون نهایی', 21),
76 ('1018374068', '09178924349', N'لنا', N'مرومی', N'فیلیم', '2023-01-06 09:40:55', N'آزمون نهایی', 29),
77 ('2013274069', '09178924350', N'رضا', N'محسنی', N'کارشناسی', '2023-01-06 12:40:05', N'آزمون نهایی', 80),
78 ('3018324070', '09178924351', N'مهران', N'رستمگاری', N'کارشناسی ارشد', '2023-01-06 15:40:15', N'آزمون نهایی', 10),

```

ساخت جدول شرکت کنندگان بر اساس جدول ثبت نامی:

```
49  -- در صورتی که جدولی با این نام وجود نداشت، آن را بساز --
50  IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Participant')
51  CREATE TABLE Participant          -- جدول شرکت کنندگان
52  (
53      National_id nchar(10) PRIMARY KEY,    -- کد ملی شرکت کنندگان
54      Mobile_number char(11) NOT NULL,      -- شماره تلفن همراه شرکت کنندگان
55      First_name nvarchar(30) NOT NULL,     -- نام شرکت کنندگان
56      Last_name nvarchar(30) NOT NULL,     -- نام خانوادگی شرکت کنندگان
57      Degree nvarchar(20) NULL,             -- مدرک شرکت کنندگان
58      Register_time datetime NOT NULL,     -- زمان ثبت نام شرکت کنندگان
59      Exam_id nvarchar(50) NOT NULL,
60      Exam_score int not null,
61  )
62  GO

98  INSERT INTO Participant
99  SELECT I.*
100 FROM Registeredpeople I
101 INNER JOIN (
102     SELECT TOP 15 National_id, MIN(Register_time) AS Min_Register
103     FROM Registeredpeople
104     WHERE Degree in (N'دپلم', N'کارشناسی', N'کارشناسی ارشد')
105     GROUP BY National_id
106 ) AS S
107 ON I.National_id = S.National_id
108 GO
```

ویرایش جدول شرکت کنندگان و اضافه کردن دو کلید خارجی:

```
110 INSERT INTO Participant
111 SELECT TOP 20 * FROM Registered.people
112 ORDER BY Register_time ASC;
113 GO
114
115 ALTER TABLE Participant
116 ADD FOREIGN KEY (National_id) REFERENCES Registered.people(National_id);
117 GO
118
119 ALTER TABLE Participant
120 ADD FOREIGN KEY (Exam_id) REFERENCES Final_exam;
121 GO
```

ساخت جدول تمرین‌ها و درج داده در آن:

```
126 -- از صورتی که جدولی با این نام وجود نداشته، آن را بساز
127 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Homework')
128 CREATE TABLE Homework -- جدول تمرین
129 (
130     Homework_id nvarchar(50) primary key, -- شماره تمرین
131     Homework_number tinyint not null, -- شماره تمرین
132     Pahse_release_time datetime null, -- زمان انتشار تمرین
133     Phase_submission_deadline datetime not null, -- مهلت ارسال تمرین
134     Project_subject nvarchar(20) not null, -- موضوع تمرین
135 )
136 Go
137
138 INSERT INTO Homework
139 VALUES
140 (N'کار با متغیر', 1, '2023-04-01 00:00:00', '2023-04-07 00:00:00', N'کار با متغیر'),
141 (N'کار تابع', 2, '2023-04-07 00:00:00', '2023-04-14 00:00:00', N'کار تابع'),
142 (N'تابع بازگشتی', 3, '2023-04-14 00:00:00', '2023-04-21 00:00:00', N'تابع بازگشتی'),
143 (N'ساختار داده', 4, '2023-04-21 00:00:00', '2023-04-28 00:00:00', N'ساختار داده');
144 GO
```

ساخت جدول پروژه و درج داده در آن:

```
146 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
147 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Project')
148 -- ایجاد جدول پروژه
149 create table Project
150 (
151     Phase_id nvarchar(50) primary key, -- شناسه پروژه
152     Phase_nom tinyint not null, -- شماره فاز پروژه
153     Phase_release_time datetime null, -- زمان ریلیز فاز پروژه
154     Phase_submission_deadline datetime not null, -- مهلت ارسال فاز پروژه
155     Project_subject nvarchar(50) not null, -- موضوع پروژه
156 )
157 Go
158
159 -- درج داده در جدول پروژه
160 Insert Into Project
161 Values
162 (N'الگوریتم مقدماتی', 1, '2023-06-01 00:00:00', '2023-06-10 00:00:00', N'فاز یک پروژه'),
163 (N'الگوریتم پیشرفته', 2, '2023-06-10 00:00:00', '2023-06-20 00:00:00', N'فاز دو پروژه')
164 Go
```

ساخت جدول رئیس بوتکمپ:

```
166 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
167 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Head_bootcamp')
168 -- ایجاد جدول رئیس بوت کمپ
169 CREATE TABLE Head_bootcamp -- جدول رئیس بوتکمپ
170 (
171     Head_id nchar(10) PRIMARY KEY, -- کد رئیس بوتکمپ
172     Mobile_number char(11) NOT NULL, -- شماره تلفن همراه رئیس بوتکمپ
173     First_name nvarchar(30) NOT NULL, -- نام رئیس بوتکمپ
174     Last_name nvarchar(30) NOT NULL, -- نام خانوادگی رئیس بوتکمپ
175     Degree nvarchar(20) NULL, -- مدرک رئیس بوتکمپ
176 )
177 GO
178
179 INSERT INTO Head_bootcamp
180 VALUES
181 ('1273694880', '09138094457', N'شایان', N'صالحی', N'دکتری')
182 GO
```

ساخت جدول شرکت های صاحب قرارداد:

```
184 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
185 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Company')
186 CREATE TABLE Company -- جدول شرکتها
187 (
188     Company_Id nchar(5) PRIMARY KEY, -- کد شرکت
189     Phone_number char(11) NOT NULL, -- شماره تلفن ثابت شرکت
190     [Name] nvarchar(50) NOT NULL, -- نام شرکت
191     Work_type nvarchar(30) NULL, -- نوع کار شرکت
192     Field nvarchar(30) NULL, -- زمینه کار شرکت
193     Salary nchar(20) NULL, -- حقوق پیشنهادی شرکت
194     Head_Id nchar(10) NOT NULL,
195     FOREIGN KEY (Head_Id) REFERENCES Head_bootcamp
196 )
197 GO
198
199 INSERT INTO Company
200 VALUES
201 ('14891', '02123192488', N'همکاران سیستم', N'حضور', N'نرم افزار', '12,000,000', '1273694880'),
202 ('31252', '02123423461', N'فناوران نوین', N'مجازی', N'تحلیل داده', '15,000,000', '1273694880'),
203 ('51626', '02151717173', N'خودران سیستم', N'مجازی', N'پایگاه داده', '13,500,000', '1273694880'),
204 ('37353', '02148235226', N'زینا الگوریتم', N'حضور', N'طراحی الگوریتم', '0,000,000', '1273694880'),
205 ('36261', '02124261465', N'نرم نشان', N'حضور', N'نرم افزار', '10,000,000', '1273694880'),
206 ('86377', '02142456169', N'خردسازان اندیشه', N'حضور', N'هوش مصنوعی', '15,360,000', '1273694880');
207 GO
```


ساخت جدول منتورها:

```

209  -- در صورتی که جدولی با این نام وجود نداشت، آن را بساز.
210  IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Mentors')
211  -- ایجاد جدول منتور ها.
212  create table Mentors
213  (
214      Mentor_id nchar(10) primary key,          -- کد ملی مربی
215      [Mobile_number] char(11) NOT NULL,
216      First_name nvarchar(30) NOT NULL,          -- نام مربی
217      Last_name nvarchar(30) NOT NULL,          -- نام خانوادگی مربی
218      Degree nvarchar(20) NULL,                 -- مدرک مربی
219      Project_id nvarchar(50) null,             -- شناسه پروژه
220      HW_id nvarchar(50) null,                  -- شناسه تمرین
221      Exam_id nvarchar(50) null,                -- شناسه آزمون نهایی
222      Head_id nchar(10) not null,
223      FOREIGN KEY (project_id) REFERENCES Project,
224      FOREIGN KEY (Exam_id) REFERENCES Final_exam,
225      FOREIGN KEY (HW_id) REFERENCES Homework,
226      FOREIGN KEY (Head_id) REFERENCES Head_bootcamp
227  )
228  GO

229  INSERT INTO Mentors
230  VALUES
231  (
232      ('1294523781', '09123316655', N'محمد', N'امین', N'دکتری', null, null, N'آزمون نهایی', '1273694880'),
233      ('1294462762', '09183778646', N'رمضان', N'دکتری', null, N'آخرین یک', null, '1273694880'),
234      ('1384442181', '09198816655', N'احمد', N'احمدی', N'دکتری', null, N'تمرین دو', null, '1273694880'),
235      ('1284522581', '09223327655', N'رضا', N'غابیش', N'کارشناسی ارشد', null, N'تمرین سه', null, '1273694880'),
236      ('1314563781', '09333399655', N'حمید', N'صابی', N'دکتری', null, N'تمرین چهار', null, '1273694880'),
237      ('1324577781', '09883316344', N'علی', N'رجایی', N'دکتری', N'فاز یک پروژه', null, null, '1273694880'),
238      ('1334523991', '09393316231', N'سیا', N'آقای', N'کارشناسی ارشد', N'فاز دو پروژه', null, null, '1273694880'),
239      ('1224888781', '09383316781', N'فران', N'مختی', N'دکتری', N'فاز یک پروژه', null, null, '1273694880'),
240      ('1214613781', '09023316694', N'آلا', N'متوسلی', N'دکتری', N'فاز دو پروژه', null, null, '1273694880'),
241      ('1194588781', '09013316593', N'مریم', N'فاقد اهمیت', N'دکتری', null, null, N'آزمون نهایی', '1273694880')
242  )
243  GO

```

ساخت جدول سابمیت‌های فردی:

```

244 -- در صورتی که جدولی با این نام وجود نداشته آن را بساز
245 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Individual_submissions_hw')
246 -- ایجاد جدول سابمیت‌های فردی
247 create table Individual_submissions_hw
248 (
249     Homework_id nvarchar(50) not null, -- شماره تمرین
250     Submission_time datetime not null, -- زمان (تاریخ + ساعت) ارسال
251     Submission_score int not null, -- نمره ارسال
252     Submitter_id nchar(10) not null, -- کد ملی فرد ارسال‌کننده
253     Constraint Submission_id_hw primary key (Submitter_id, Submission_time), -- شماره فرد ارسال‌کننده
254     FOREIGN KEY (Submitter_id) REFERENCES Participant,
255 )
256 Go

```

```

257 -- درج داده در جدول سابمیت‌های فردی
258 Insert Into Individual_submissions_hw
259 values
260 (N'تمرین یک', '2023-05-02 16:20:11', 12, '0352465468'),
261 (N'تمرین یک', '2023-05-03 17:45:23', 13, '0352465468'),
262 (N'تمرین یک', '2023-05-04 20:20:45', 13, '0352465468'),
263 (N'تمرین یک', '2023-05-06 03:20:32', 15, '0352465468'),
264 (N'تمرین یک', '2023-05-06 16:20:15', 18, '0352465468'),
265 (N'تمرین یک', '2023-05-03 12:11:23', 5, '0784688419'),
266 (N'تمرین یک', '2023-05-04 12:33:23', 10, '0784688419'),
267 (N'تمرین یک', '2023-05-04 02:01:23', 18, '0784688419'),
268 (N'تمرین یک', '2023-05-05 12:22:23', 22, '0784688419'),
269 (N'تمرین یک', '2023-05-06 14:11:23', 24, '0784688419'),
270 (N'تمرین یک', '2023-05-02 23:16:38', 17, '0318960291'),
271 (N'تمرین یک', '2023-05-02 23:56:01', 25, '0318960291'),
272 (N'تمرین دو', '2023-05-08 12:16:46', 15, '0352465468'),
273 (N'تمرین دو', '2023-05-10 15:46:09', 25, '0352465468'),
274 (N'تمرین دو', '2023-05-09 11:21:42', 18, '0352465468'),
275 (N'تمرین دو', '2023-05-11 16:29:35', 19, '0352465468'),
276 (N'تمرین دو', '2023-05-11 19:23:47', 25, '0352465468'),
277 (N'تمرین دو', '2023-05-10 15:46:09', 25, '0318960291'),
278 (N'تمرین سه', '2023-05-15 20:47:12', 18, '0352465468'),
279 (N'تمرین سه', '2023-05-15 19:37:12', 17, '0352465468'),
280 (N'تمرین سه', '2023-05-15 11:24:14', 25, '0352465468'),
281 (N'تمرین سه', '2023-05-17 18:24:33', 20, '0973274052'),
282 (N'تمرین سه', '2023-05-20 13:21:42', 16, '0784688419'),
283 (N'تمرین سه', '2023-05-20 23:21:32', 23, '0784688419'),
284 (N'تمرین چهار', '2023-05-21 17:22:54', 21, '0352465468'),
285 (N'تمرین چهار', '2023-05-22 17:11:54', 23, '0352465468'),
286 (N'تمرین چهار', '2023-05-26 15:36:24', 24, '0352465468'),
287 (N'تمرین چهار', '2023-05-28 14:56:17', 25, '0352465468'),

```

ساخت جدول سابمیت‌های نهایی تمرین بر اساس سابمیت‌های فردی:

```
285 -- فردی که جدول نهایی را می‌سازد. اگر از قبل وجود داشته باشد، آن را حذف می‌کند.
286 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Final_submissions_hw')
287 -- ایجاد جدول سابمیت‌های نهایی.
288 create table Final_submissions_hw
289 (
290     Homework_id nvarchar(50) not null, -- شماره تمرین
291     Submission_time datetime not null, -- زمان (تاریخ + ساعت) ارسال
292     Submission_score int not null, -- نمره ارسال
293     Submitter_id nchar(10) not null, -- کد ملی فرد ارسال‌کننده
294     Constraint Final_Submission_id_hw primary key (Submitter_id, Submission_time), -- شماره فرد ارسال‌کننده
295 )
296 GO
297
298 INSERT INTO Final_submissions_hw
299 SELECT I.*
300 FROM Individual_submissions_hw I
301 INNER JOIN (
302     SELECT Submitter_id, MAX(Submission_score) AS Max_Score
303     FROM Individual_submissions_hw
304     GROUP BY Submitter_id
305 ) AS S
306 ON I.Submitter_id = S.Submitter_id AND I.Submission_score = S.Max_Score
307 GO
308
309 ALTER TABLE Final_submissions_hw
310 ADD FOREIGN KEY (Submitter_id, Submission_time) REFERENCES Individual_submissions_hw (Submitter_id, Submission_time);
311 GO
312
313 ALTER TABLE Final_submissions_hw
314 ADD FOREIGN KEY (Homework_id) REFERENCES Homework (Homework_id);
315 GO
```


ساخت جدول ارسال‌های فردی پروژه:

```

327 -- بررسی می‌کنیم که جدولی با این نام وجود نداشته باشد یا نه
328 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Individual_submissions_project')
329 -- ایجاد جدول مستطیت هائی فردی پروژه در هر شی
330 create table Individual_submissions_project
331 (
332     Phase_id nvarchar(50) not null, -- شماره تمرین
333     Submission_time datetime not null, -- زمان (تاریخ + ساعت) ارسال
334     Submission_score int not null, -- نمره ارسال
335     Submitter_id nchar(10) not null, -- کد ملی فرد ارسال‌کننده
336     Constraint Submission_id_project primary key(Submitter_id, Submission_time), -- شناسه فرد ارسال‌کننده
337     FOREIGN KEY (Submitter_id) REFERENCES Participant,
338 )
339 Go
340
341 Insert Into Individual_submissions_project
342 values
343 (N'فاز یک پروژه', '2023-06-02 16:20:11', 50, '0352465468'),
344 (N'فاز یک پروژه', '2023-06-03 17:45:23', 55, '0352465468'),
345 (N'فاز یک پروژه', '2023-06-04 20:20:45', 55, '0352465468'),
346 (N'فاز یک پروژه', '2023-06-06 03:20:32', 60, '0352465468'),
347 (N'فاز یک پروژه', '2023-06-06 16:20:15', 70, '0352465468'),
348 (N'فاز یک پروژه', '2023-06-03 12:11:23', 20, '0784688419'),
349 (N'فاز یک پروژه', '2023-06-04 12:33:23', 65, '0784688419'),
350 (N'فاز یک پروژه', '2023-06-04 02:01:23', 75, '0784688419'),
351 (N'فاز یک پروژه', '2023-06-05 12:22:23', 89, '0784688419'),
352 (N'فاز یک پروژه', '2023-06-06 14:11:23', 95, '0784688419'),
353 (N'فاز یک پروژه', '2023-06-02 23:16:38', 70, '0318960291'),
354 (N'فاز یک پروژه', '2023-06-02 23:56:01', 100, '0318960291'),
355 (N'فاز یک پروژه', '2023-06-08 12:16:46', 70, '0352465468'),
356 (N'فاز یک پروژه', '2023-06-10 15:46:09', 100, '0352465468'),
357 (N'فاز یک پروژه', '2023-06-09 11:21:42', 75, '0352465468'),
358 (N'فاز یک پروژه', '2023-06-11 16:29:35', 75, '0352465468'),
359 (N'فاز یک پروژه', '2023-06-11 19:23:47', 100, '0352465468').

```

ساخت جدول ارسال‌های نهایی پروژه بر اساس ارسال‌های فردی:

```

377 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
378 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Final_submissions_project')
379 -- ایجاد جدول Submission های نهایی پروژه
380 create table Final_submissions_project
381 (
382     Phase_id nvarchar(50) not null, -- شماره آخرین
383     Submission_time datetime not null, -- زمان (تاریخ + ساعت) ارسال
384     Submission_score int not null, -- نمره ارسال
385     Submitter_id nchar(10) not null, -- کد ملی فرد ارسال‌کننده
386     Constraint final_Submission_id_project primary key (Submitter_id, Submission_time), -- کلید فرد ارسال‌کننده
387 )
388 GO
389
390 INSERT INTO Final_submissions_project
391 SELECT I.*
392 FROM Individual_submissions_project I
393 INNER JOIN (
394     SELECT Submitter_id, MAX(Submission_score) AS Max_Score
395     FROM Individual_submissions_project
396     GROUP BY Submitter_id
397 ) AS S
398 ON I.Submitter_id = S.Submitter_id AND I.Submission_score = S.Max_Score
399 GO
400
401 ALTER TABLE Final_submissions_project
402 ADD FOREIGN KEY (Submitter_id, Submission_time) REFERENCES Individual_submissions_project (Submitter_id, Submission_time);
403 GO
404
405 ALTER TABLE Final_submissions_project
406 ADD FOREIGN KEY (Phase_id) REFERENCES Project (Phase_id);
407 GO
408

```

ساخت جدول اساتید:

```

409 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
410 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Teacher')
411 -- ایجاد جدول اساتید
412 CREATE TABLE Teacher
413 (
414     Teacher_id nchar(10) PRIMARY KEY, -- کد ملی اساتید
415     Mobile_number char(11) NOT NULL, -- شماره تلفن همراه اساتید
416     First_name nvarchar(30) NOT NULL, -- نام اساتید
417     Last_name nvarchar(30) NOT NULL, -- نام خانوادگی اساتید
418     Degree nvarchar(20) NULL, -- مدرک اساتید
419     Email varchar(50) NULL, -- ایمیل اساتید
420 )
421 GO
422
423 INSERT INTO Teacher
424 VALUES
425 ('1414523974', '09163316655', N'آرمین', N'رضایی', N'دکتری', 'armin.betoche@gmail.com'),
426 ('1444493781', '09283776646', N'علیرضا', N'محمدزاده', N'دکتری', 'ali.khastam@gmail.com'),
427 ('1884433181', '09298828855', N'عاطفه', N'نیازمند', N'دکتری', 'atni.nemikhan@gmail.com'),
428 ('1584522491', '09823335655', N'سارا', N'اجمالی', N'کارشناسی ارشد', 'sara.hey@gmail.com'),
429 ('1384563755', '09533411655', N'نیره', N'حیدری', N'دکتری', 'nayare.movafagh@gmail.com')
430 GO
431

```

ساخت جدول کلاس:

```
432 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
433 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Class')
434 -- ایجاد جدول کلاس
435 CREATE TABLE Class
436 (
437     Class_id nchar(5) PRIMARY KEY,
438     Teacher_id nchar(10) NOT NULL,
439     Beginning_time time NOT NULL,
440     End_time time NOT NULL,
441     [Subject] nvarchar(50) NOT NULL,
442     FOREIGN KEY (Teacher_id) REFERENCES Teacher
443 )
444
445 GO
446
447 INSERT INTO Class
448 VALUES
449     ('21221', '1414523974', '16:00:00', '18:00:00', N'الگوریتم خه'),
450     ('21741', '1444493781', '14:00:00', '15:30:00', N'پیدامازی اساب بازی'),
451     ('31241', '1884433181', '22:00:00', '23:30:30', N'نیامای اولیه'),
452     ('62622', '1584522491', '10:00:00', '12:30:00', N'مقدمات ساختار'),
453     ('72262', '1384563755', '17:00:00', '19:00:00', N'هوش ساختگی')
454
455 GO
```

ساخت جدول ارتباط بین کلاس و شرکت کننده با دو کلید خارجی:

```
456 -- در صورتی که جدولی با این نام وجود نداشته، آن را بساز
457 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Class_participant')
458 CREATE TABLE Class_participant
459 (
460     Class_id nchar(5) NOT NULL,
461     Participant_id nchar(10) NOT NULL,
462     Constraint Class_participant_id primary key(Class_id, Participant_id),
463     FOREIGN KEY (Participant_id) REFERENCES Participant(National_id),
464     FOREIGN KEY (Class_id) REFERENCES Class
465 )
466 GO
467
468 INSERT INTO Class_participant
469 VALUES
470     ('21221', '0318960291'),
471     ('21221', '0784688419'),
472     ('21221', '0973274050'),
473     ('21221', '1052465470'),
474     ('21221', '1093274058'),
475     ('21741', '0352465468'),
476     ('21741', '4062865461'),
477     ('21741', '0318960291'),
478     ('31241', '0973274050'),
479     ('62622', '7063274050'),
480     ('72262', '6063274055')
481 GO
482
```

ساخت جدول قبول شدگان بر اساس نمره نهایی آنها:

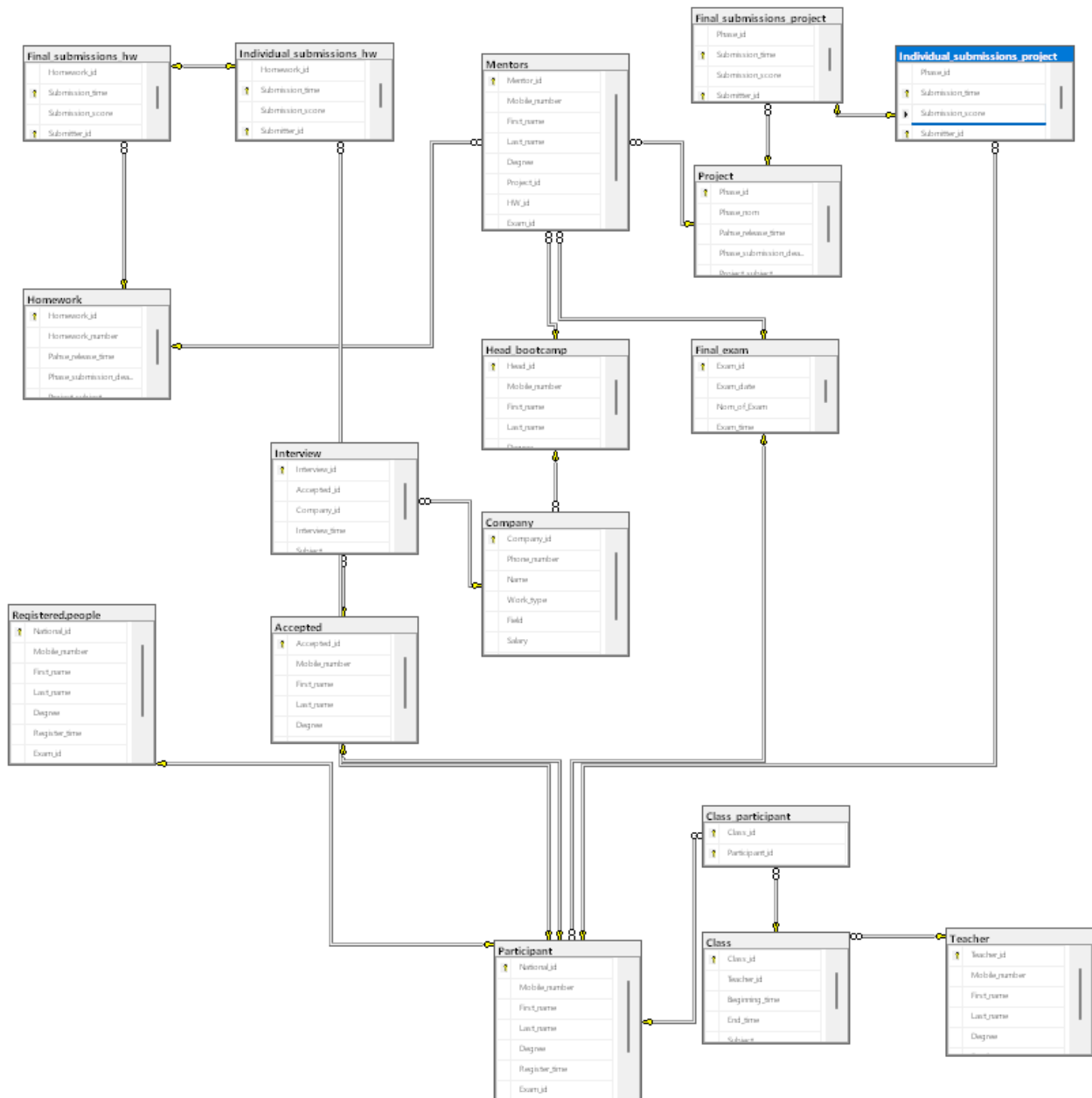
```
483 -- در صورتی که جدولی با این نام وجود نداشت، آن را بساز
484 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Accepted')
485 -- ساخت جدول قبول شدگان براساس نمرات کل تمرین و پروژه و آزمون نهایی
486 CREATE TABLE Accepted -- جدول قبول شدگان
487 (
488     Accepted_id nchar(10) PRIMARY KEY, -- کد ملی قبول شدگان
489     Mobile_number char(11) NOT NULL, -- شماره تلفن همراه قبول شدگان
490     First_name nvarchar(30) NOT NULL, -- نام قبول شدگان
491     Last_name nvarchar(30) NOT NULL, -- نام خانوادگی قبول شدگان
492     Degree nvarchar(20) NULL, -- مدرک قبول شدگان
493     All_score int not null,
494 )
495 GO
```

```
496 -- INSERT INTO Accepted
497 SELECT TOP 10 Participant.National_id AS participant_id, Mobile_number, First_name, Last_name, Degree, (TOTAL(Final_exam_score.Final_exam_score, 0) + TOTAL(Final_HW
498 FROM Participant
499 FULL OUTER JOIN
500 (SELECT National_id AS participant_id_1, Exam_score AS Final_exam_score FROM Participant) AS Final_exam_score
501 ON Participant.National_id = Final_exam_score.participant_id_1
502 FULL OUTER JOIN
503 (SELECT HW_scores.participant_id AS participant_id_2, SUM(HW_scores.score) AS HW_score
504 FROM (SELECT Final_submissions.hw.submitter_id AS participant_id, Final_submissions.hw.homework_id, MAX(Final_submissions.hw.Submission_score) AS score
505 FROM Individual_submissions_hw
506 JOIN Final_submissions_hw
507 ON Individual_submissions_hw.submitter_id = Final_submissions.hw.submitter_id
508 GROUP BY Final_submissions.hw.homework_id, Final_submissions.hw.submitter_id) AS HW_scores
509 GROUP BY HW_scores.participant_id) AS Final_HW_score
510 ON Final_exam_score.participant_id_1 = Final_HW_score.participant_id_2
511 FULL OUTER JOIN
512 (SELECT project_scores.participant_id, SUM(project_scores.score) AS project_score
513 FROM (SELECT Final_submissions_project.submitter_id AS participant_id, Final_submissions_project.Phase_id, MAX(Final_submissions_project.Submission_score) AS score
514 FROM Final_submissions_project
515 GROUP BY Final_submissions_project.Phase_id, Final_submissions_project.submitter_id) AS project_scores
516 GROUP BY project_scores.participant_id) AS Final_project_score
517 ON Final_HW_score.participant_id_2 = Final_project_score.participant_id
518 GROUP BY all_scores.participant_id
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

ساخت جدول مصاحبه پذیرفته شدگان:

```
529 -- بررسی که جدولی با این نام وجود داشته آن را بساز
530 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'Interview')
531 -- ساخت جدول مصاحبه پذیرفته شدگان
532 CREATE TABLE Interview
533 (
534     Interview_id nchar(10) PRIMARY KEY,
535     Accepted_id nchar(10) NOT NULL,
536     Company_id nchar(5) NOT NULL,
537     Interview_time datetime NOT NULL,
538     [Subject] nvarchar(50) NOT NULL
539     FOREIGN KEY (Accepted_id) REFERENCES Accepted(Accepted_id),
540     FOREIGN KEY (Company_id) REFERENCES Company(Company_id)
541 )
542 GO
543
544 INSERT INTO Interview
545 VALUES
546     ('8289935211', '0352465468', '14891', '2023-08-01 16:00:00', N'موش مصنوعی'),
547     ('8255621372', '0318960291', '31252', '2023-08-02 16:00:00', N'الگوریتم پیشرفته'),
548     ('8282345157', '0784688419', '37353', '2023-08-01 18:00:00', N'ساختار داده'),
549     ('8264256149', '0973274055', '86377', '2023-08-03 18:00:00', N'نیاز مالی'),
550     ('8242169241', '1294274044', '14891', '2023-08-01 18:00:00', N'پایگاه ماشین')
551 GO
552
```

در نهایت برای دیاگرام تولید شده پس از ساخت این جداول خواهیم داشت:



فصل ۶

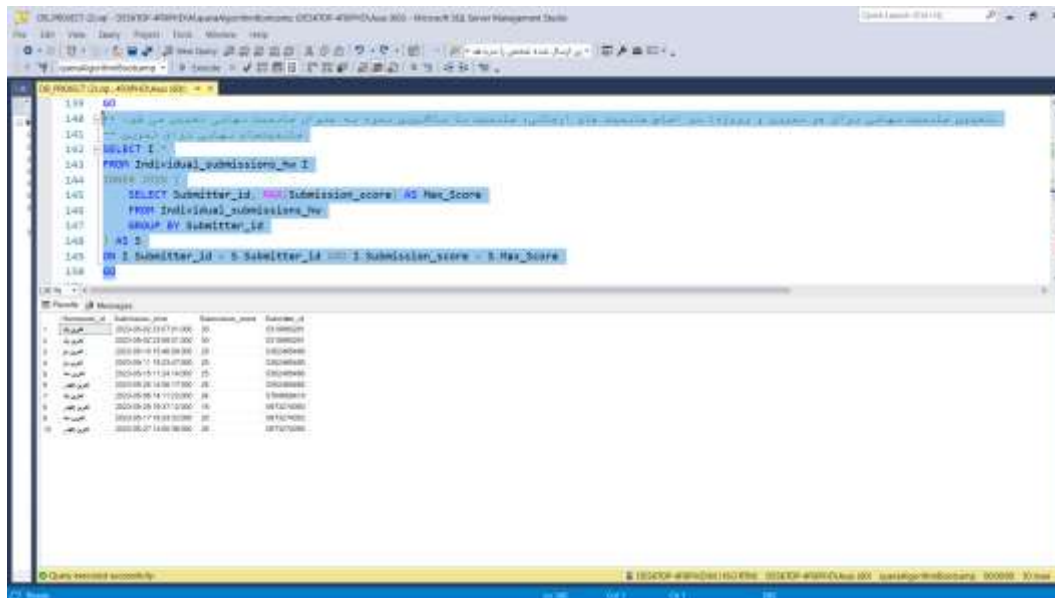
لیست عملیات و گزارشات پروژه

۳-۳. عملیات و گزارشات پروژه

- درج شرکت کننده: مشخصات شرکت کننده، مانند نام، سن، مدرک تحصیلی و زمان ثبت نام ورودی خواهند بود.
- بررسی مجاز بودن شرکت کننده: بر اساس شرایط ثبت نام، بررسی می شود که آیا شرکت کننده مجاز به شرکت در بوتکمپ است یا خیر.
- درج تمرین: مشخصات تمرین مانند محتوا، مهلت ارسال، منتور طراح، نمره و غیره ورودی خواهند بود.
- درج پروژه: مشخصات پروژه مانند محتوا، مهلت ارسال برای هر فاز، منتور طراح، نمره و غیره ورودی خواهند بود.
- ثبت سابمیت تمرین: این عملیات شامل ثبت تمرین هایی است که شرکت کننده ارسال کرده است.
- ثبت سابمیت پروژه: این عملیات شامل ثبت پروژه هایی است که شرکت کننده ارسال کرده است.
- تعیین سابمیت نهایی برای هر تمرین و پروژه: بر اساس سابمیت های ارسالی، سابمیت با بالاترین نمره به عنوان سابمیت نهایی تعیین می شود.

تمامی عملیات های بالا در فصل قبلی با استفاده از عملیات INSERT INTO table_name انجام شده و کد sql آن نیز قرار داده شده است.

برای سابمیت تمرین:



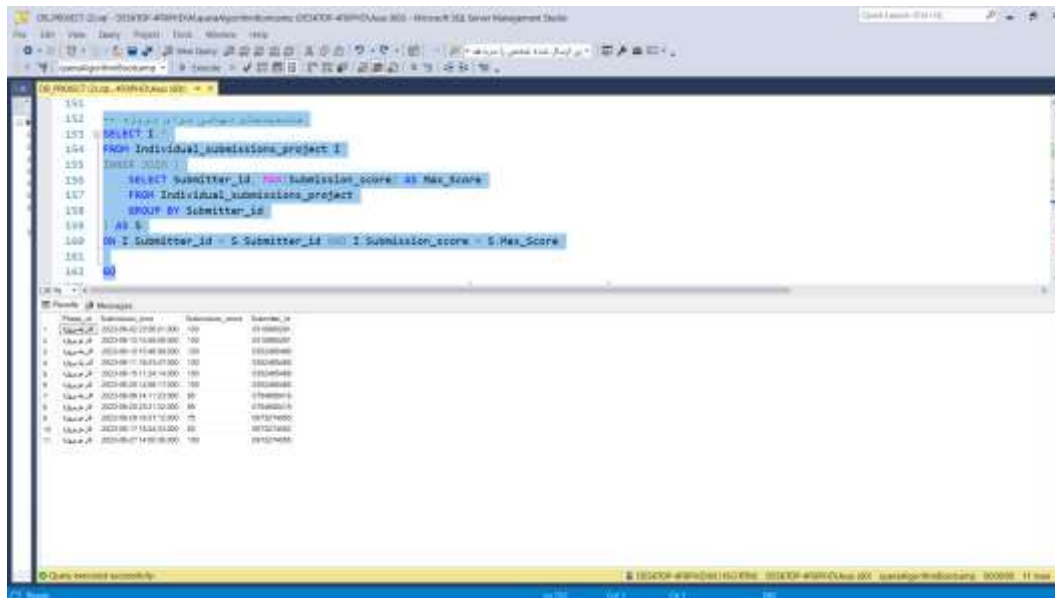
که کوئری آن به صورت زیر است:

--
تعیین سابمیت نهایی برای هر تمرین و پروژه: بر اساس سابمیت های ارسالی، سابمیت با بالاترین نمره به عنوان سابمیت نهایی تعیین می شود.

--
سابمیت های نهایی برای تمرین

```
SELECT I.*
FROM Individual_submissions_hw I
INNER JOIN (
    SELECT Submitter_id, MAX(Submission_score) AS Max_Score
    FROM Individual_submissions_hw
    GROUP BY Submitter_id
) AS S
ON I.Submitter_id = S.Submitter_id AND I.Submission_score = S.Max_Score
GO
```

و برای سابمیت نهایی پروژه:



که کوئری آن به صورت زیر است:

-- سابمیت های نهایی برای پروژه --

SELECT I.*

FROM Individual_submissions_project I

INNER JOIN (

SELECT Submitter_id, MAX(Submission_score) AS Max_Score

FROM Individual_submissions_project

GROUP BY Submitter_id

) AS S

ON I.Submitter_id = S.Submitter_id AND I.Submission_score = S.Max_Score

GO

- درج آزمون نهایی: مشخصات آزمون نهایی مانند تاریخ اجرا و نمره ورودی خواهند بود:

The screenshot shows a SQL Server Enterprise Manager interface. The query window displays the following SQL query:

```

148 ON s.Submitter_id = s.Submitter_id AND s.Submission_score = s.Max_Score
149
150
151
152
153
154 SELECT *, Participant.Exam_score FROM Final_exam
155
156 FULL OUTER JOIN Participant
157 ON Final_exam.Exam_id = Participant.Exam_id
158
159

```

The results pane shows a table with the following columns: Exam_id, Exam_score, Submitter_id, Submission_score, Exam_name, Exam_date, Exam_time, Exam_place, Exam_score, Exam_score. The table contains 20 rows of data, including exam details for various subjects like 'معماری', 'معماری-2', 'معماری-3', 'معماری-4', 'معماری-5', 'معماری-6', 'معماری-7', 'معماری-8', 'معماری-9', 'معماری-10', 'معماری-11', 'معماری-12', 'معماری-13', 'معماری-14', 'معماری-15', 'معماری-16', 'معماری-17', 'معماری-18', 'معماری-19', 'معماری-20'.

که کوئری آن به صورت زیر است:

- درج آزمون نهایی: مشخصات آزمون نهایی مانند تاریخ اجرا و نمره ورودی خواهند بود - -

SELECT *, Participant.Exam_score FROM Final_exam

FULL OUTER JOIN Participant

ON Final_exam.Exam_id = Participant.Exam_id

GO

- ساخت جدولی با استفاده از سلکت و جوین‌های تو در تو که کل مشخصات شرکت‌کننده‌ها به همراه نمرات نهایی تمرین، پروژه و آزمون نهایی‌شان را نشان می‌دهد:

participant_id	mobile_number	first_name	last_name	degree	register_time	final_exam_score	hw_scores
1	9876543210	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
2	9876543211	سید	علی	کارشناسی	2020-01-01 10:00:00	75	75
3	9876543212	علی	محمد	کارشناسی	2020-01-01 10:00:00	85	85
4	9876543213	علی	محمد	کارشناسی	2020-01-01 10:00:00	70	70
5	9876543214	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
6	9876543215	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
7	9876543216	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
8	9876543217	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
9	9876543218	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
10	9876543219	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
11	9876543220	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
12	9876543221	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
13	9876543222	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
14	9876543223	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
15	9876543224	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
16	9876543225	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
17	9876543226	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
18	9876543227	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75
19	9876543228	علی	محمد	کارشناسی	2020-01-01 10:00:00	80	80
20	9876543229	علی	محمد	کارشناسی	2020-01-01 10:00:00	75	75

کوئری آن به صورت زیر است:

جدول نمرات کل تمام شرکت کنندگان --

```

SELECT Participant.National_id AS
participant_id, Mobile_number, First_name, Last_name, Degree, Register_time,
(ISNULL(final_exam_score.final_exam_score, 0) + ISNULL(final_HW_score.HWs_score, 0) +
ISNULL(final_project_score.project_score, 0)) AS all_scores
FROM Participant
FULL OUTER JOIN
(SELECT National_id AS participant_id_1, Exam_score AS final_exam_score FROM Participant)
AS final_exam_score
ON Participant.National_id = final_exam_score.participant_id_1
FULL OUTER JOIN
(SELECT HW_scores.participant_id AS participant_id_2, SUM(HW_scores.score) AS
HWs_score
from (SELECT Final_submissions_hw.Submitter_id AS participant_id,
Final_submissions_hw.Homework_id, MAX(Final_submissions_hw.Submission_score) AS score
FROM Final_submissions_hw

```

```

GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores
GROUP BY HW_scores.participant_id) AS final_HW_score
ON final_exam_score.participant_id_1 = final_HW_score.participant_id_2
FULL OUTER join
(SELECT project_scores.participant_id, SUM(project_scores.score) AS project_score
from (SELECT Final_submissions_project.Submitter_id AS participant_id ,
Final_submissions_project.Phase_id, MAX(Final_submissions_project.Submission_score) AS
score
FROM Final_submissions_project
GROUP BY Final_submissions_project.Phase_id, Final_submissions_project.Submitter_id) AS
project_scores
GROUP BY project_scores.participant_id) AS final_project_score
ON final_HW_score.participant_id_2 = final_project_score.participant_id;

GO

```

تا به اینجا عملیات‌های گفته شده به صورت درج و قرار دادن داده‌های جدید بوده، می‌توانیم علاوه بر این عملیات‌ها، کارهای دیگری همانند انتخاب، تابع و روال را انجام دهیم. عملیات‌های پایین

```

168 -- انتخاب ۵ نفر اول تمرین بر اساس نمره داده شده به آنها
169 SELECT TOP 5 Participant.National_id AS participant_id,
170 SUM(ISNULL(HW_scores.score, 0)) AS HW_score
171 FROM Participant FULL OUTER JOIN
172 (SELECT Final_submissions_hw.Submitter_id AS participant_id, Final_submissions_hw.Homework_id,
173 MAX(Final_submissions_hw.Submission_score) AS score,
174 MAX(Final_submissions_hw.Submission_time) AS Submission_time
175 FROM Final_submissions_hw
176 JOIN Individual_submissions_hw
177 ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
178 GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS HW_scores
179 ON HW_scores.participant_id = Participant.National_id
180 GROUP BY HW_scores.score, Participant.National_id
181 ORDER BY HW_score DESC
182
183 GO

```

participant_id	HW_score
0352495458	75
0318960291	30
0319960291	25
0373274022	25
0784888419	24

- انتخاب ۵ نفر اول تمرین بر اساس نمره داده شده به آنها:

که کوئری آن به صورت زیر است:

-- انتخاب ۵ نفر اول تمرین بر اساس نمره داده شده به آنها

```

SELECT TOP 5 Participant.National_id AS participant_id,
SUM(ISNULL(HW_scores.score, 0)) AS HW_score
FROM Participant FULL OUTER JOIN
(SELECT      Final_submissions_hw.Submitter_id      AS
Final_submissions_hw.Homework_id,
MAX(Final_submissions_hw.Submission_score) AS score,
MAX(Final_submissions_hw.Submission_time) AS Submission_time
FROM Final_submissions_hw
JOIN Individual_submissions_hw
ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores

```

```
ON HW_scores.participant_id = Participant.National_id
GROUP BY HW_scores.score, Participant.National_id
ORDER BY HW_score DESC;
```

GO

- انتخاب ۵ نفر اول تمرین بر اساس زمان ارسال آنها:

```

184
185 -- انتخاب ۵ نفر اول تمرین بر اساس زمان ارسال آنها --
186 SELECT TOP 5 Participant.National_id AS participant_id,
187 MAX(HW_scores.Submission_time) AS Submission_time
188 FROM Participant FULL OUTER JOIN
189 (SELECT Final_submissions_hw.Submitter_id AS participant_id, Final_submissions_hw.Homework_id,
190 MAX(Final_submissions_hw.Submission_score) AS score,
191 MAX(Final_submissions_hw.Submission_time) AS Submission_time
192 FROM Final_submissions_hw
193 JOIN Individual_submissions_hw
194 ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
195 GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS HW_scores
196 ON HW_scores.participant_id = Participant.National_id
197 WHERE HW_scores.Submission_time IS NOT NULL
198 GROUP BY HW_scores.Submission_time, Participant.National_id
199 ORDER BY HW_scores.Submission_time
200
201 GO

```

	participant_id	Submission_time
1	0318960291	2023-05-02 23:59:01.000
2	0794608419	2023-05-06 14:11:23.000
3	0318960291	2023-05-10 15:46:06.000
4	0302489488	2023-05-11 19:23:47.000
5	0302489488	2023-05-15 11:24:14.000

که کوئری آن به صورت زیر است:

انتخاب ۵ نفر اول تمرین بر اساس زمان ارسال آنها - -

```

SELECT TOP 5 Participant.National_id AS participant_id,
MAX(HW_scores.Submission_time) AS Submission_time
FROM Participant FULL OUTER JOIN
(SELECT
    Final_submissions_hw.Submitter_id AS
    participant_id,
Final_submissions_hw.Homework_id,
MAX(Final_submissions_hw.Submission_score) AS score,
MAX(Final_submissions_hw.Submission_time) AS Submission_time

```



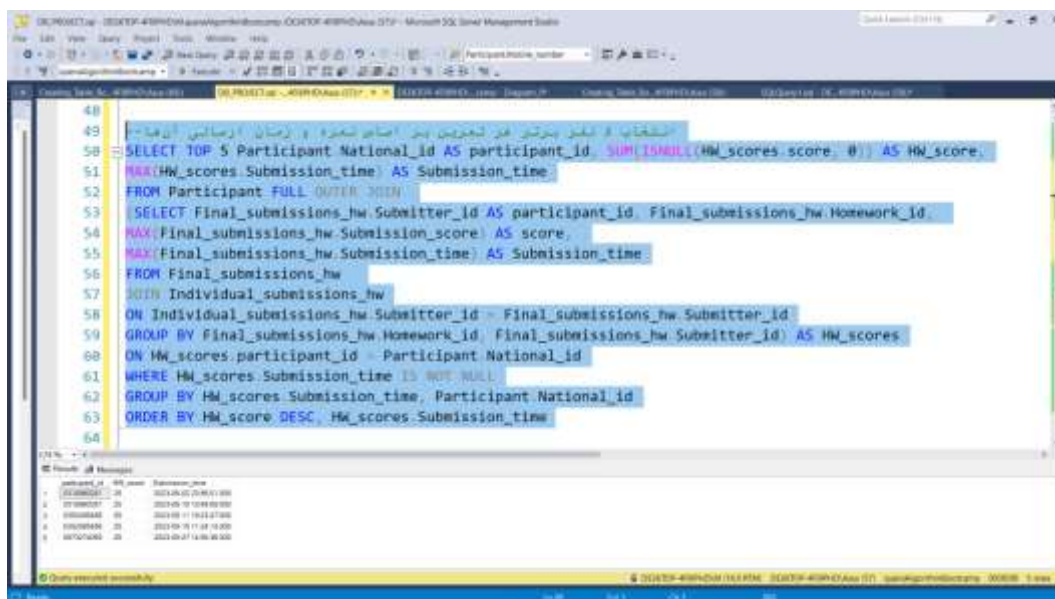
```

FROM Final_submissions_hw
JOIN Individual_submissions_hw
ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores
ON HW_scores.participant_id = Participant.National_id
WHERE HW_scores.Submission_time IS NOT NULL
GROUP BY HW_scores.Submission_time, Participant.National_id
ORDER BY HW_scores.Submission_time;

GO

```

- انتخاب ۵ نفر برتر هر تمرین بر اساس نمره و زمان ارسالی آنها:



که کوئری آن به صورت زیر است:

انتخاب ۵ نفر برتر هر تمرین بر اساس نمره و زمان ارسالی آنها - -

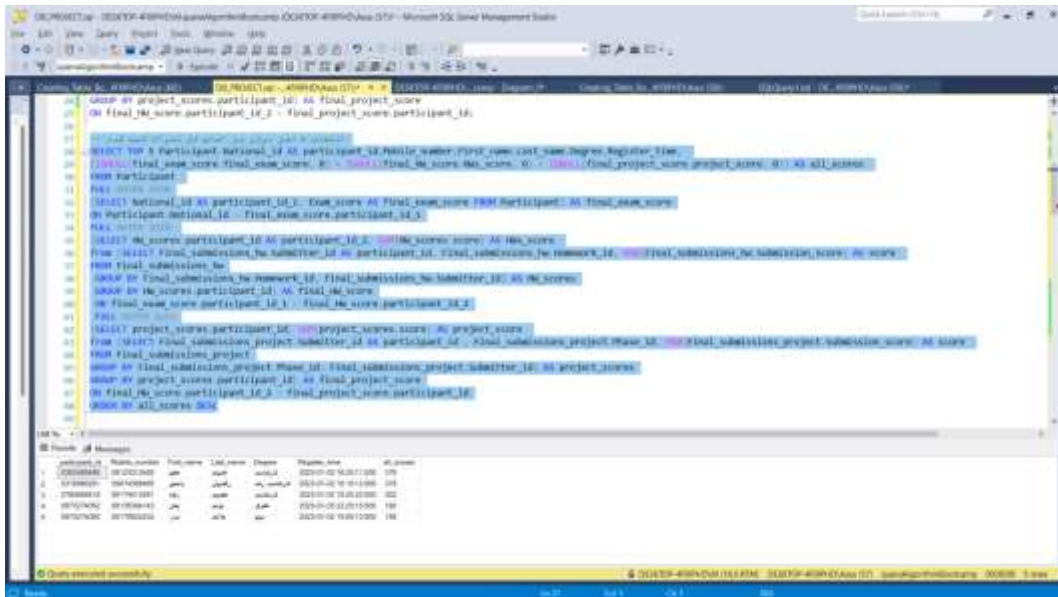
```

SELECT TOP 5 Participant.National_id AS participant_id, SUM(ISNULL(HW_scores.score, 0))
AS HW_score,

```

```
MAX(HW_scores.Submission_time) AS Submission_time
FROM Participant FULL OUTER JOIN
(SELECT      Final_submissions_hw.Submitter_id      AS      participant_id,
Final_submissions_hw.Homework_id,
MAX(Final_submissions_hw.Submission_score) AS score,
MAX(Final_submissions_hw.Submission_time) AS Submission_time
FROM Final_submissions_hw
JOIN Individual_submissions_hw
ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores
ON HW_scores.participant_id = Participant.National_id
WHERE HW_scores.Submission_time IS NOT NULL
GROUP BY HW_scores.Submission_time, Participant.National_id
ORDER BY HW_score DESC, HW_scores.Submission_time;
GO
```

- انتخاب ۵ نفر برتر بر اساس کل نمرات کسب شده:



که کوئری آن به صورت زیر است:

انتخاب ۵ نفر برتر بر اساس کل نمرات کسب شده --

SELECT TOP 5 Participant.National_id **AS**

participant_id,Mobile_number,First_name,Last_name,Degree,Register_time,

(**ISNULL**(final_exam_score.final_exam_score, 0) + **ISNULL**(final_HW_score.HWs_score, 0) +

ISNULL(final_project_score.project_score, 0)) **AS** all_scores

FROM Participant

FULL OUTER JOIN

(**SELECT** National_id **AS** participant_id_1, Exam_score **AS** final_exam_score **FROM** Participant)

AS final_exam_score

ON Participant.National_id = final_exam_score.participant_id_1

FULL OUTER JOIN

(**SELECT** HW_scores.participant_id **AS** participant_id_2, **SUM**(HW_scores.score) **AS**

HWs_score

from (**SELECT** Final_submissions_hw.Submitter_id **AS** participant_id,

Final_submissions_hw.Homework_id, **MAX**(Final_submissions_hw.Submission_score) **AS** score

FROM Final_submissions_hw

```

GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores
GROUP BY HW_scores.participant_id) AS final_HW_score
ON final_exam_score.participant_id_1 = final_HW_score.participant_id_2
FULL OUTER join
(SELECT project_scores.participant_id, SUM(project_scores.score) AS project_score
from (SELECT Final_submissions_project.Submitter_id AS participant_id ,
Final_submissions_project.Phase_id, MAX(Final_submissions_project.Submission_score) AS
score
FROM Final_submissions_project
GROUP BY Final_submissions_project.Phase_id, Final_submissions_project.Submitter_id) AS
project_scores
GROUP BY project_scores.participant_id) AS final_project_score
ON final_HW_score.participant_id_2 = final_project_score.participant_id
ORDER BY all_scores DESC;
GO

```

- تعریف تابعی که به عنوان ورودی اسم شرکت کننده را گرفته و در خروجی تعداد تمرین ارسال شده شخص را می دهد:

```

-- تعریف تابعی که به عنوان ورودی اسم شرکت کننده را گرفته و در خروجی تعداد تمرین ارسال شده شخص را می دهد:
CREATE PROCEDURE Get_exam_Scores
    @participant_id NVARCHAR(100)
AS
BEGIN
    SELECT Participant.National_id AS participant_id_1, COUNT(HW_scores.score) AS HW_count
    FROM Participant
    FULL JOIN
    (SELECT Final_submissions_hw.Submitter_id AS participant_id, Final_submissions_hw.Homework_id, MAX(Final_submissions_hw.Submission_score) AS score
    FROM Final_submissions_hw
    JOIN HW_scores ON Final_submissions_hw.Submitter_id = HW_scores.participant_id
    GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS HW_scores
    ON HW_scores.participant_id = Participant.National_id
    WHERE Participant.National_id = @participant_id
    GROUP BY Participant.National_id
END
EXEC Get_exam_Scores @participant_id = '123456789'

```

که کوئری آن به صورت زیر است:

• تعریف تابعی که به عنوان ورودی اسم شرکت کننده را گرفته و در خروجی تعداد تمرین ارسال شده شخص را می دهد - -

```
CREATE PROCEDURE Get_count_Scores
@participant_id nchar(10)
AS
BEGIN
SELECT Participant.National_id AS participant_id_1, COUNT(ISNULL(HW_scores.score, 0))
AS HWs_count
FROM Participant
FULL OUTER JOIN
(SELECT Final_submissions_hw.Submitter_id AS participant_id,
Final_submissions_hw.Homework_id,
MAX(Final_submissions_hw.Submission_score) AS score
FROM Individual_submissions_hw
JOIN Final_submissions_hw
ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS
HW_scores
ON HW_scores.participant_id = Participant.National_id
WHERE Participant.National_id = @participant_id
GROUP BY Participant.National_id
END
GO
EXEC Get_count_Scores @participant_id = '0318960291';
GO
```

- تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و نمرات تمرین اول و دوم را به عنوان خروجی می دهد:

```

76 DROP PROCEDURE IF EXISTS Get_HW_1_2_Scores;
77 GO
78 -- تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و نمرات تمرین اول و دوم را به عنوان خروجی می دهد
79 CREATE PROCEDURE Get_HW_1_2_Scores
80 @participant_id nchar(10)
81 AS
82 BEGIN
83     SELECT Participant.National_id AS participant_id_1, SUM(ISNULL(HW_scores.score, 0)) AS score
84     FROM Participant
85     FULL OUTER JOIN
86     (SELECT Final_submissions_hw.Submitter_id AS participant_id, Final_submissions_hw.Homework_id,
87      MAX(Final_submissions_hw.Submission_score) AS score
88      FROM Individual_submissions_hw
89      JOIN Final_submissions_hw
90      ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
91      GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id) AS HW_scores
92     ON HW_scores.participant_id = Participant.National_id
93     WHERE Participant.National_id = @participant_id and (Homework_id = 'تمرین یک' or Homework_id = 'تمرین دو')
94     GROUP BY Participant.National_id
95 END
96 GO
97 EXEC Get_HW_1_2_Scores @participant_id = '0123456789';
98 GO

```

که کوئری آن به صورت زیر است:

• تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و نمرات تمرین اول و دوم را به عنوان خروجی می دهد - -

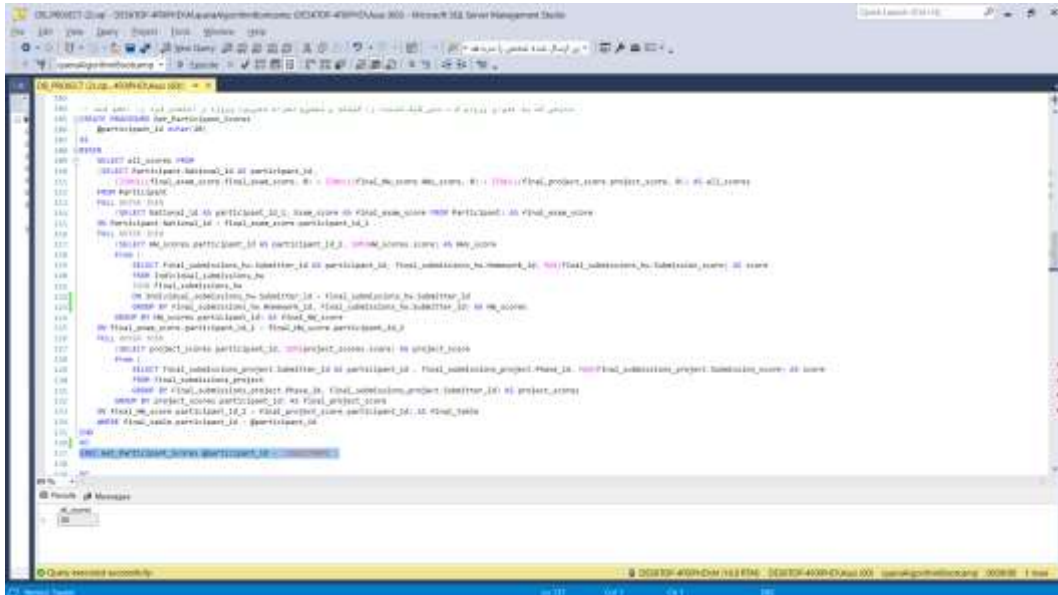
```

CREATE PROCEDURE Get_HW_1_2_Scores
@participant_id nchar(10)
AS
BEGIN
    SELECT Participant.National_id AS participant_id_1, SUM(ISNULL(HW_scores.score,
0)) AS score
    FROM Participant
    FULL OUTER JOIN
    (SELECT Final_submissions_hw.Submitter_id AS participant_id,
Final_submissions_hw.Homework_id,
MAX(Final_submissions_hw.Submission_score) AS score
    FROM Individual_submissions_hw
    JOIN Final_submissions_hw
    ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
    GROUP BY Final_submissions_hw.Homework_id, Final_submissions_hw.Submitter_id)
AS HW_scores
    ON HW_scores.participant_id = Participant.National_id
    WHERE Participant.National_id = @participant_id and (Homework_id = 'تمرین یک' or
Homework_id = 'تمرین دو')
    GROUP BY Participant.National_id
END
GO

```

```
EXEC Get_HW_1_2_Scores @participant_id = '0318960291';  
GO
```

- تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و مجموع نمرات تمرین، پروژه و امتحان فرد را اعلام کند.



که کوئری آن به صورت زیر است:

- تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و مجموع نمرات تمرین، پروژه و امتحان فرد را اعلام کند --

```
CREATE PROCEDURE Get_Participant_Scores  
@participant_id nchar(10)  
AS  
BEGIN  
    SELECT all_scores FROM  
    (SELECT Participant.National_id AS participant_id,  
    (ISNULL(final_exam_score.final_exam_score, 0) + ISNULL(final_HW_score.HWs_score,  
0) + ISNULL(final_project_score.project_score, 0)) AS all_scores  
FROM Participant  
FULL OUTER JOIN  
    (SELECT National_id AS participant_id_1, Exam_score AS final_exam_score FROM  
Participant) AS final_exam_score  
ON Participant.National_id = final_exam_score.participant_id_1  
FULL OUTER JOIN  
    (SELECT HW_scores.participant_id AS participant_id_2, SUM(HW_scores.score) AS  
HWs_score  
from (  
    SELECT Final_submissions_hw.Submitter_id AS participant_id,  
Final_submissions_hw.Homework_id, MAX(Final_submissions_hw.Submission_score) AS score  
FROM Individual_submissions_hw  
JOIN Final_submissions_hw  
ON Individual_submissions_hw.Submitter_id = Final_submissions_hw.Submitter_id
```

```

GROUP BY Final_submissions_hw.Homework_id,
Final_submissions_hw.Submitter_id) AS HW_scores
GROUP BY HW_scores.participant_id) AS final_HW_score
ON final_exam_score.participant_id_1 = final_HW_score.participant_id_2
FULL OUTER JOIN
(SELECT project_scores.participant_id, SUM(project_scores.score) AS project_score
from (
SELECT Final_submissions_project.Submitter_id AS participant_id ,
Final_submissions_project.Phase_id, MAX(Final_submissions_project.Submission_score) AS
score
FROM Final_submissions_project
GROUP BY Final_submissions_project.Phase_id,
Final_submissions_project.Submitter_id) AS project_scores
GROUP BY project_scores.participant_id) AS final_project_score
ON final_HW_score.participant_id_2 = final_project_score.participant_id) AS final_table
WHERE final_table.participant_id = @participant_id
END
GO
EXEC Get_Participant_Scores @participant_id = '3103274061'

GO

```

- همچنین می‌دانیم که در شرکت‌ها و نیز این بوت‌کمپ، داده‌ها باید به صورت خودکار آپدیت شوند! یعنی زمانی که فردی سابمیتی در تمرین و یا پروژه داشت، باید به صورت خودکار، جدول سابمیت‌های نهایی تمرین و پروژه نیز آپدیت شوند. و از طرفی جدول نمرات کل افراد و در ادامه جدول پذیرفته‌شدگان نیز در صورت نیاز باید به طور خودکار آپدیت شوند؛ بنابراین باید برای این ۴ قسمت تریگر نوشت که با تغییر هر قسمت به طور خودکار جداول مذکور آپدیت شوند:
- ساخت تریگر در صورتی که به سابمیت‌های فردی تمرین داده‌ای اضافه شد، جدول سابمیت‌های نهایی تمرین نیز آپدیت شود:

ساخت تریگر در صورتی که به سابمیت‌های فردی تمرین اضافه شد جدول سابمیت‌های نهایی تمرین نیز آپدیت شود.

```

CREATE TRIGGER trg_AfterInsert_HW
ON Individual_submissions_hw
AFTER INSERT AS
BEGIN
INSERT INTO Final_submissions_hw
SELECT i.*
FROM inserted i
LEFT JOIN Final_submissions_hw f
ON f.Submitter_id = i.Submitter_id AND f.Submission_time < i.Submission_time
WHERE f.Submitter_id IS NULL OR i.Submission_score > f.Submission_score

```



```
END;  
GO
```

- ساخت تریگر در صورتی که به سابمیت‌های فردی پروژه داده‌ای اضافه شد، جدول

سابمیت‌های نهایی پروژه نیز آپدیت شود:

• ساخت تریگر در صورتی که به سابمیت‌های فردی پروژه اضافه شد جدول سابمیت‌های نهایی پروژه نیز آپدیت شود --

```
CREATE TRIGGER trg_AfterInsert_Project  
ON Individual_submissions_project  
AFTER INSERT AS  
BEGIN  
    INSERT INTO Final_submissions_project  
    SELECT i.*  
    FROM inserted i  
    LEFT JOIN Final_submissions_project f  
    ON f.Submitter_id = i.Submitter_id AND f.Submission_time < i.Submission_time  
    WHERE f.Submitter_id IS NULL OR i.Submission_score > f.Submission_score  
END;  
GO
```

- ساخت تریگر در صورتی که به سابمیت‌های نهایی تمرین داده‌ای اضافه شد، جدول

پذیرفته‌شدگان در صورت تغییر آپدیت شود:

• ساخت تریگر که در صورتی که به ارسال‌های نهایی تمرین نمره‌ای اضافه شد جدول پذیرفته‌شدگان در صورت تغییر آپدیت شود

```
CREATE TRIGGER trg_Final_submissions_hw  
ON Final_submissions_hw  
AFTER INSERT  
AS  
BEGIN  
    INSERT INTO Accepted(Accepted_id, All_score)  
    SELECT i.Submitter_id, i.Submission_score  
    FROM inserted i  
    WHERE NOT EXISTS (SELECT 1 FROM Accepted a WHERE a.Accepted_id =  
i.Submitter_id)  
END  
GO
```

- ساخت تریگر در صورتی که به سابمیت‌های نهایی پروژه داده‌ای اضافه شد، جدول پذیرفته‌شدگان در صورت تغییر آپدیت شود:

ساخت تریگر که در صورتی که به ارسال‌های نهایی پروژه نمره‌ای اضافه شد جدول پذیرفته‌شدگان در صورت تغییر آپدیت شود

```
CREATE TRIGGER trg_Final_submissions_project
ON Final_submissions_project
AFTER INSERT
AS
BEGIN
INSERT INTO Accepted(Accepted_id, All_score)
SELECT i.Submitter_id, i.Submission_score
FROM inserted i
WHERE NOT EXISTS (SELECT 1 FROM Accepted a WHERE a.Accepted_id =
i.Submitter_id)
END
GO
```

همچنین در ساخت بعضی از جدوال از انتخاب بهره گرفته شده، برای مثال جدول افراد شرکت‌کننده براساس جدول افراد ثبت‌نامی ساخته می‌شود، به طوری که ۲۰ نفری که زودتر از مابقی افراد ثبت‌نام کرده باشند از جدول ثبت‌نامی انتخاب شده و جدول شرکت‌کننده را شکل می‌دهد.

• بخش امتیازی:

برای قسمت امتیازی این پروژه، از زبان برنامه‌نویسی پایتون استفاده خواهیم کرد. پایتون یکی از زبان‌های برنامه‌نویسی با کاربرد بالاست و توانایی برقراری اتصال ساده و مستقیم به هر سرور SQL را دارد، که این امکان را می‌دهد تا کوئری‌های مورد نیاز اجرا شوند.

در واقع در تجربه‌ی نزدیک به شرکت‌ها متوجه می‌شویم که برنامه‌نویس‌های BACKEND

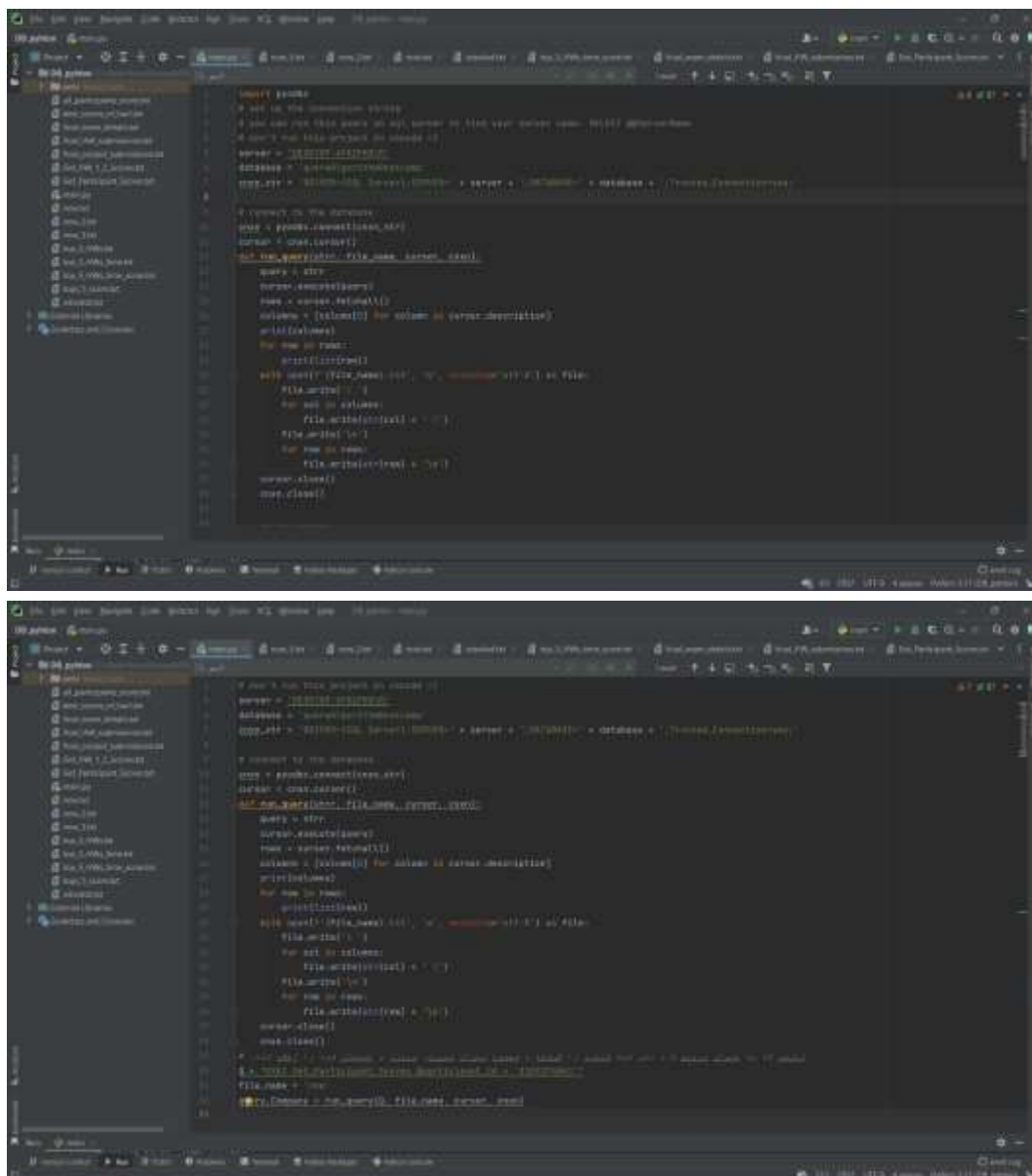
مستقیم با دیتابیس سروکار ندارند و یا حتی به علت محرمانگی داده‌ها اجازه‌ی دسترسی به دیتابیس را ندارند! اما جاهایی نیاز دارند که قسمتی از توابع دیتابیس را ران کرده و خروجی‌اش را ببینند. (برای مثال برای دیباگ)

در پروژه‌ی ما هم برای اینکه رئیس بوت‌کمپ، تی‌ای‌ها و ... بتوانند عملکرد شرکت‌کننده‌ها را بررسی کنند و یا نتیجه‌گیری از نمرات تمارین امتحان و یا حتی در قسمت ثبت نام، نیاز دارند که کوئری‌ها را اجرا کنند. (که باز هم می‌گوییم به علت محرمانگی در شرکت‌ها، آن‌ها فقط توابع را صدا می‌زنند و ران می‌کنند و خروجی را مشاهده می‌کنند.)

اما ما در اینجا علاوه بر اینکه توابع را ران خواهیم کرد، کوئری‌ها را هم در پایتون اجرا می‌کنیم تا از صحت عملکرد آن مطمئن شویم.

همچنین چون فقط در شرکت‌ها نتیجه‌را به صرف دیدن نمی‌خواهند و برایشان مهم است که نتیجه را ذخیره هم داشته باشند تا بتوانند آن را بین بقیه‌ی همکاران منتقل کنند و سند کنند، در این کد علاوه بر اینکه خروجی در کنسول چاپ خواهد شد، در فایل txt هم ذخیره خواهد شد.

مراحل کد نیز در کد کامنت شده است:



در این کد یک تابع داریم که علاوه بر `cursor`, `cnxn` و رودی‌های کوئری (که همان کوئری‌ای هست که می‌خواهیم اجرا شود) و نام فایل (نام فایلی که می‌خواهیم خروجی را در فایل `.txt` با آن نام بسازد را می‌گیرد و در نهایت خروجی‌های مذکور را می‌دهد.

کد آن نیز به صورت زیر است:

```

import pyodbc

# set up the connection string
# you can run this query on sql server to find your server name: SELECT @@ServerName
# don't run this project in vscode =)
server = 'DESKTOP-4F0IPHD\M'
database = 'queraAlgorithmBootcamp'
cnxn_str = 'DRIVER={SQL Server};SERVER=' + server + ';DATABASE=' + database +
';Trusted_Connection=yes;'

# connect to the database
cnxn = pyodbc.connect(cnxn_str)
cursor = cnxn.cursor()

def run_query(strr, file_name, cursor, cnxn):
    query = strr
    cursor.execute(query)
    rows = cursor.fetchall()
    columns = [column[0] for column in cursor.description]
    print(columns)
    for row in rows:
        print(list(row))
    with open(f'{file_name}.txt', 'w', encoding='utf-8') as file:
        file.write('| ')
        for col in columns:
            file.write(str(col) + ' |')
        file.write('\n')
        for row in rows:
            file.write(str(row) + '\n')
    cursor.close()
    cnxn.close()

# کند اعلام را فرد امتحان و پروژه تمرین، نمرات مجموع و گرفته را کننده شرکت ملی د ک ورودی عنوان به که تابعی
Q = "EXEC Get_Participant_Scores @participant_id = '3103274061'"
file_name = 'now'
query_Company = run_query(Q, file_name, cursor, cnxn)

```

در ادامه عملیات‌هایی که در SQL زدیم را در اینجا خروجی فایل و کنسول را بعد از ران کردن برنامه نشان می‌دهیم: (لازم به ذکر است که کوئری‌های این قسمت همگی در قسمت “لیست عملیات‌ها” قرار داده شده است).

۱. بیشترین نمره‌ی تمرین اول:

The top screenshot shows the C# code for the 'FindMaxScore' method. It iterates through a list of submissions and finds the maximum score for each submitter. The code is as follows:

```

for (int i = 0; i < submissions.Count; i++)
{
    int maxScore = 0;
    for (int j = 0; j < submissions[i].Submissions.Count; j++)
    {
        if (submissions[i].Submissions[j].Score > maxScore)
        {
            maxScore = submissions[i].Submissions[j].Score;
        }
    }
    Console.WriteLine($"Submitter {i+1} has a max score of {maxScore}");
}

```

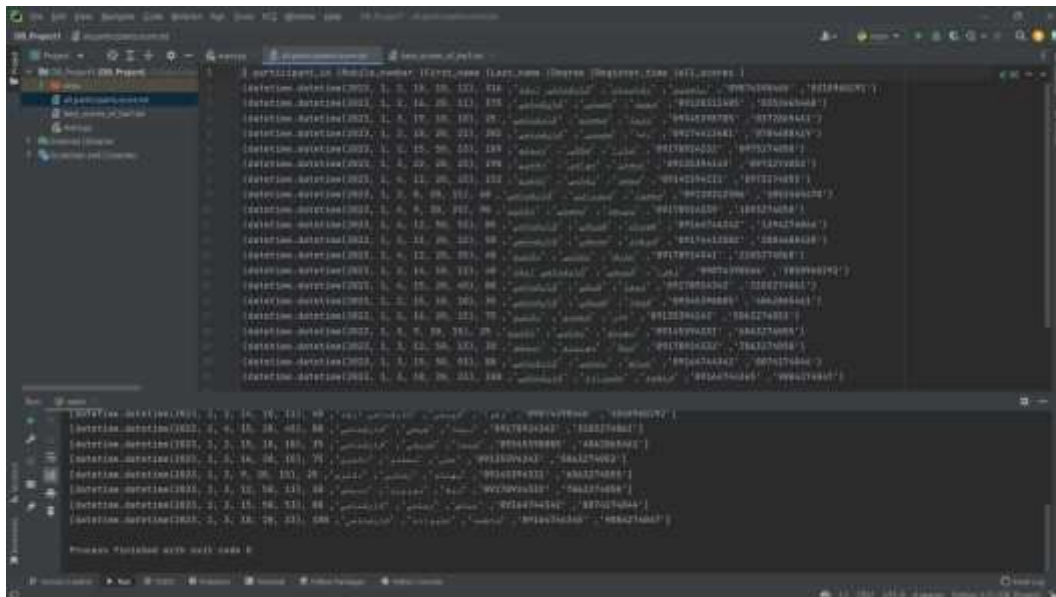
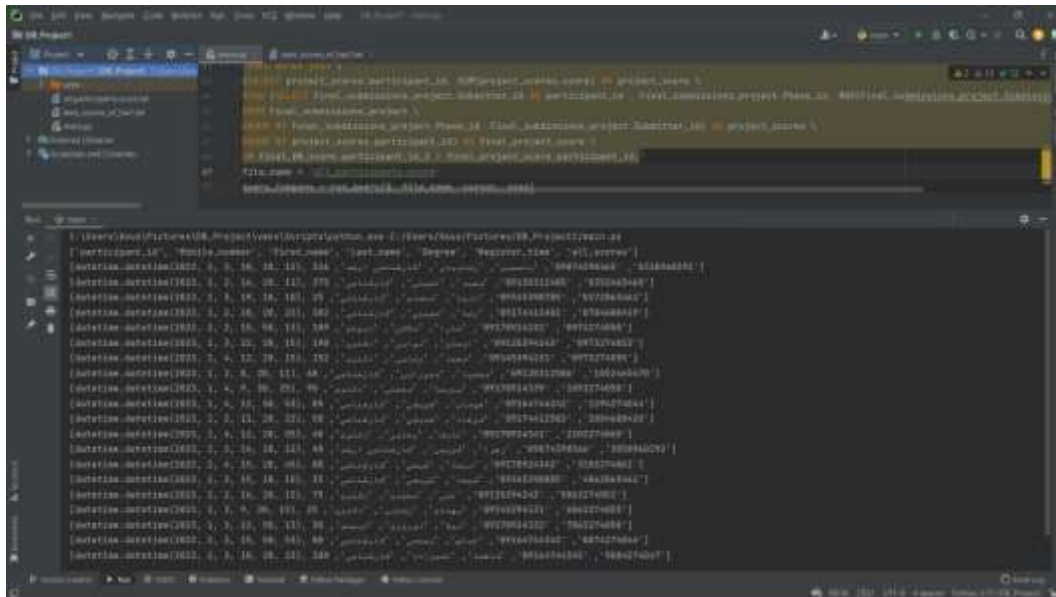
The bottom screenshot shows the application's output window. It displays the results of the query, which are the maximum scores for each submitter. The output is as follows:

```

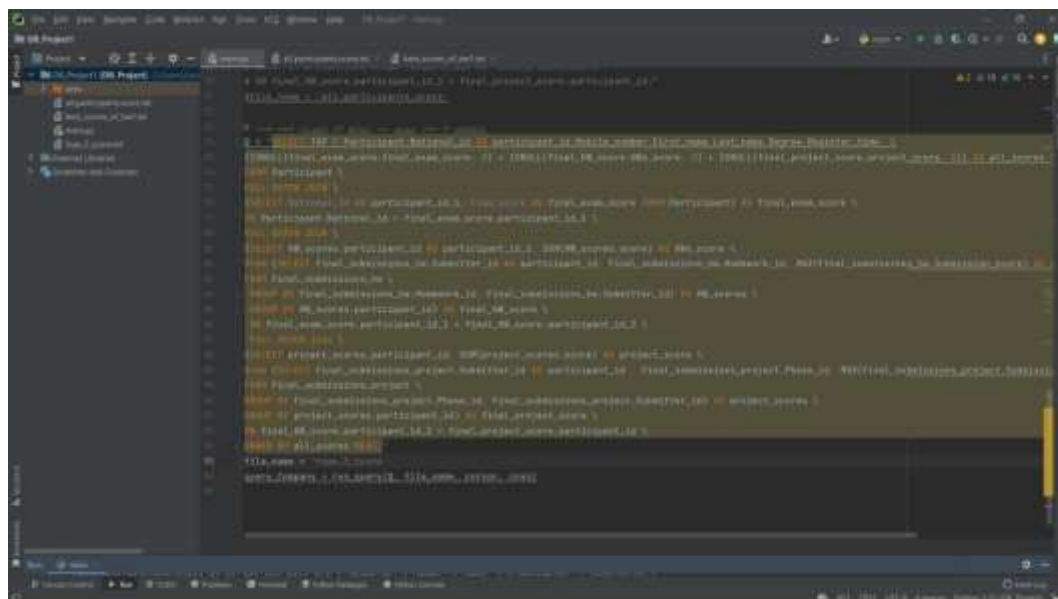
Submitter 1 has a max score of 100
Submitter 2 has a max score of 100
Submitter 3 has a max score of 100
Submitter 4 has a max score of 100
Submitter 5 has a max score of 100
Submitter 6 has a max score of 100
Submitter 7 has a max score of 100
Submitter 8 has a max score of 100
Submitter 9 has a max score of 100
Submitter 10 has a max score of 100

```

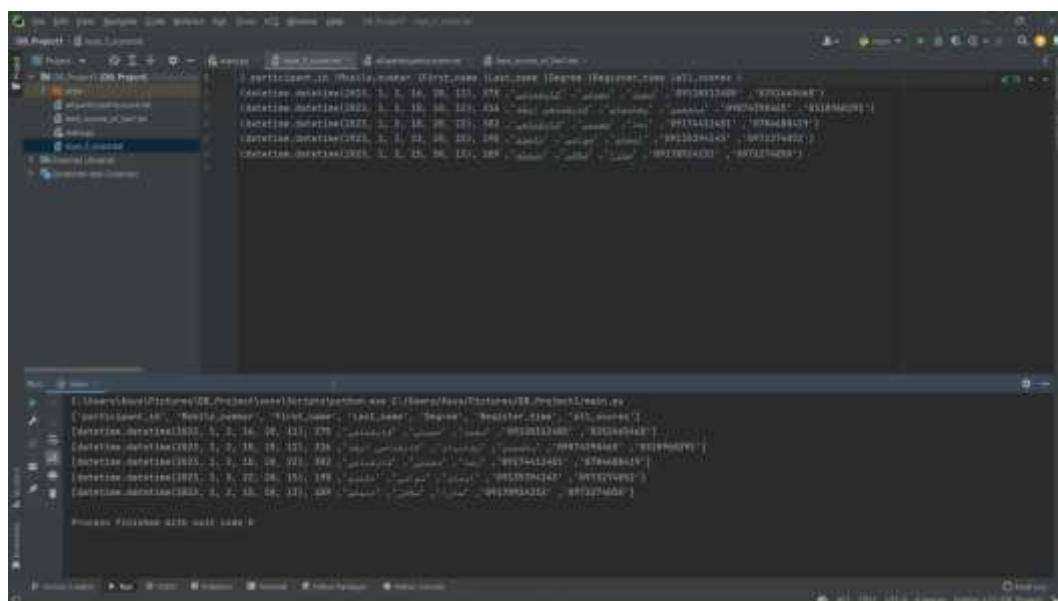
جدول نمرات کل تمام شرکت کننده‌ها:



انتخاب ۵ نفر برتر بر اساس کل نمرات کسب شده:

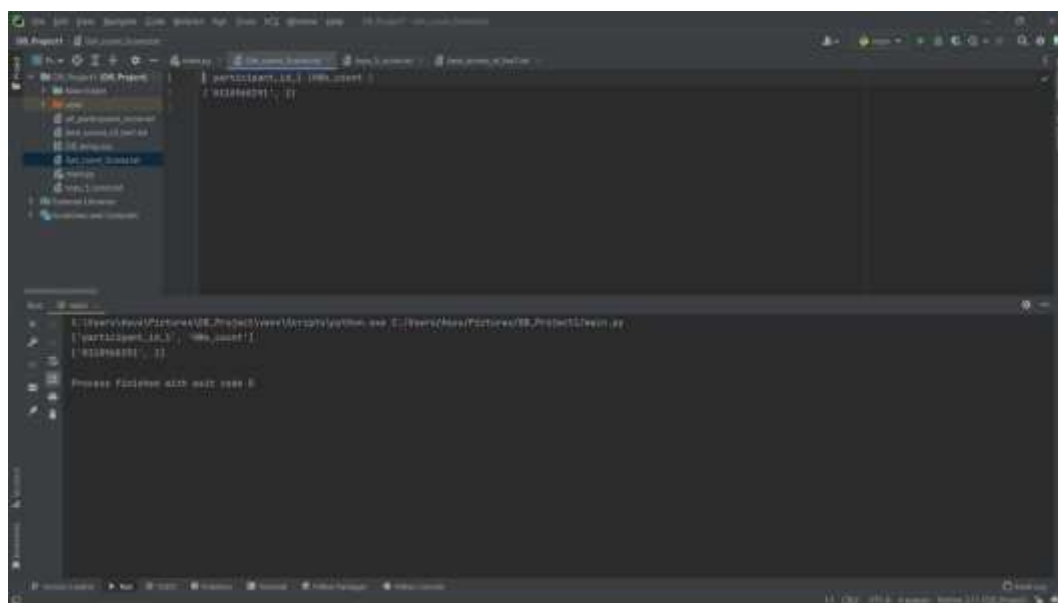
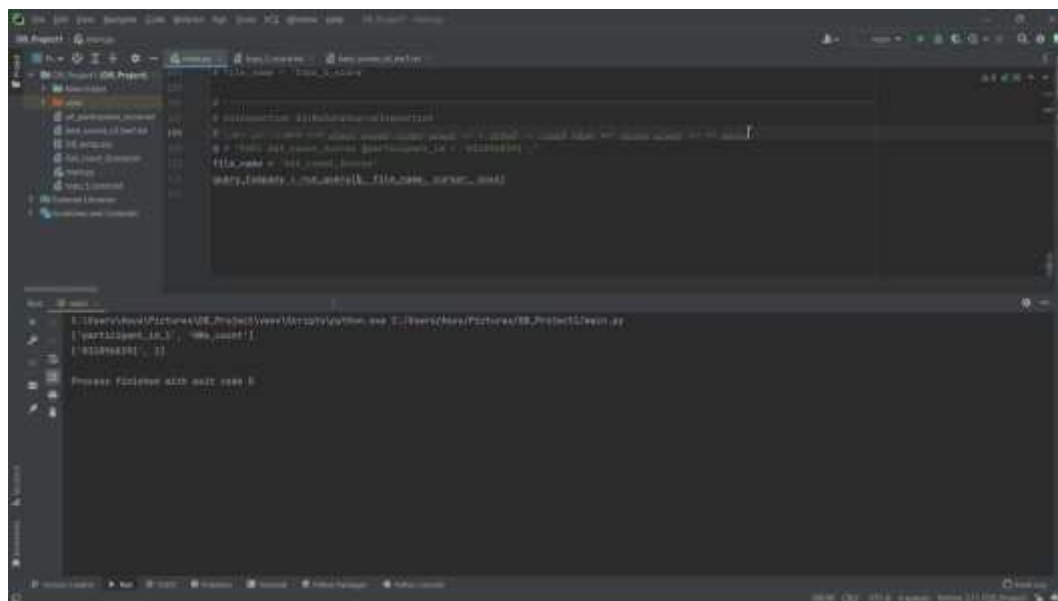


```
SELECT * FROM (
  SELECT student_id, first_name, last_name, degree_register_time,
         (SUM(score) OVER (PARTITION BY student_id ORDER BY degree_register_time)) AS total_score
  FROM student, degree_register
  WHERE student_id = degree_register_student_id
)
ORDER BY total_score DESC
FETCH FIRST 5 ROWS ONLY;
```

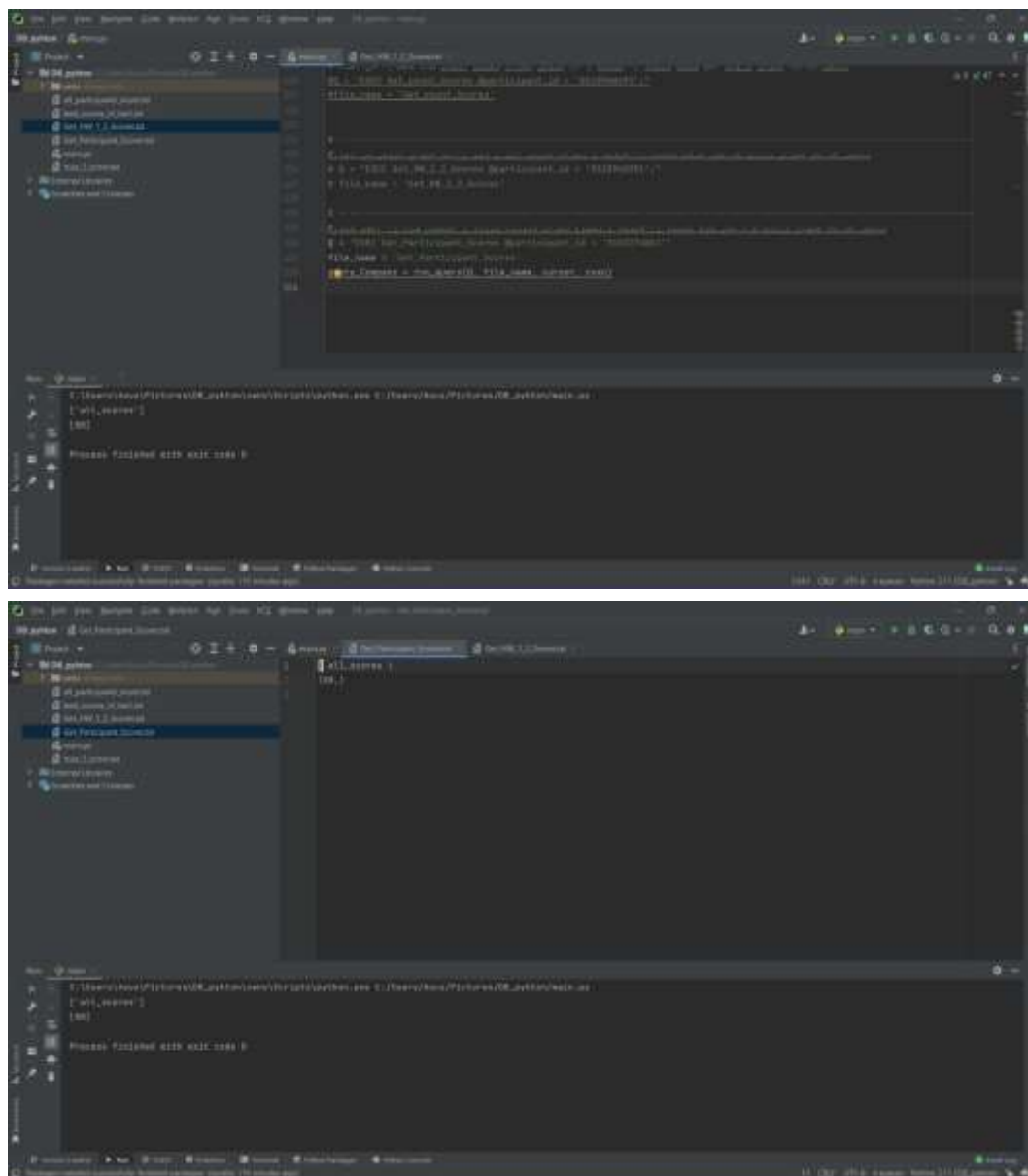


```
student_id | first_name | last_name | degree_register_time | total_score |
-----+-----+-----+-----+-----+
1000000000 | ... | ... | ... | ... |
1000000000 | ... | ... | ... | ... |
1000000000 | ... | ... | ... | ... |
1000000000 | ... | ... | ... | ... |
1000000000 | ... | ... | ... | ... |
```

تابعی که به عنوان ورودی اسم شرکت کننده را گرفته و در خروجی تعداد تمرین ارسال شده شخص را می‌دهد::

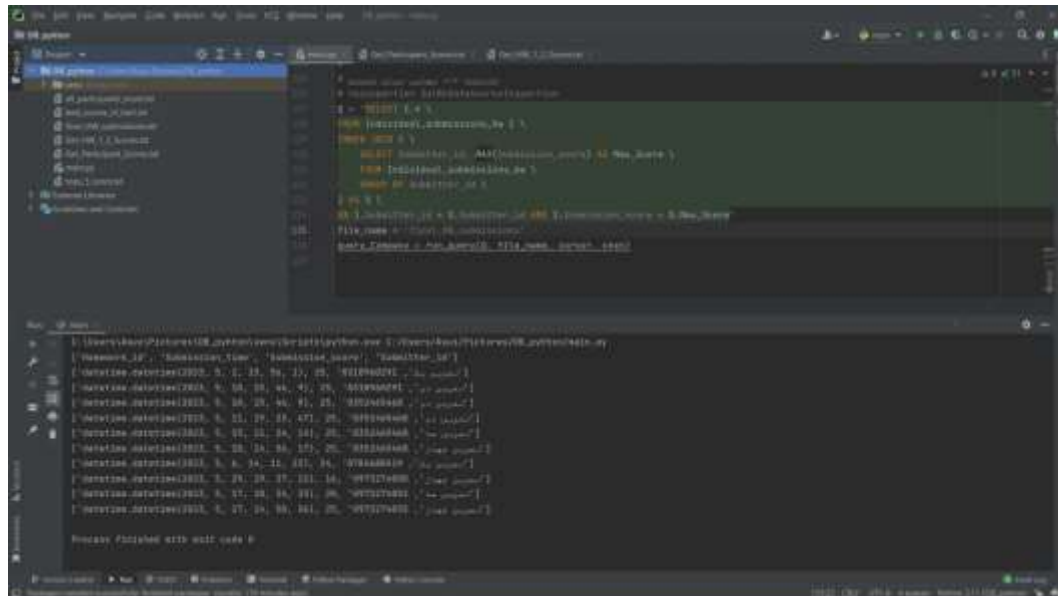


تابعی که به عنوان ورودی کد ملی شرکت کننده را گرفته و نمرات تمرین اول و دوم را به عنوان خروجی می دهد:



تعیین سابمیت نهایی برای هر تمرین و پروژه: بر اساس سابمیت های ارسالی، سابمیت بالاترین نمره به عنوان سابمیت نهایی تعیین می شود.

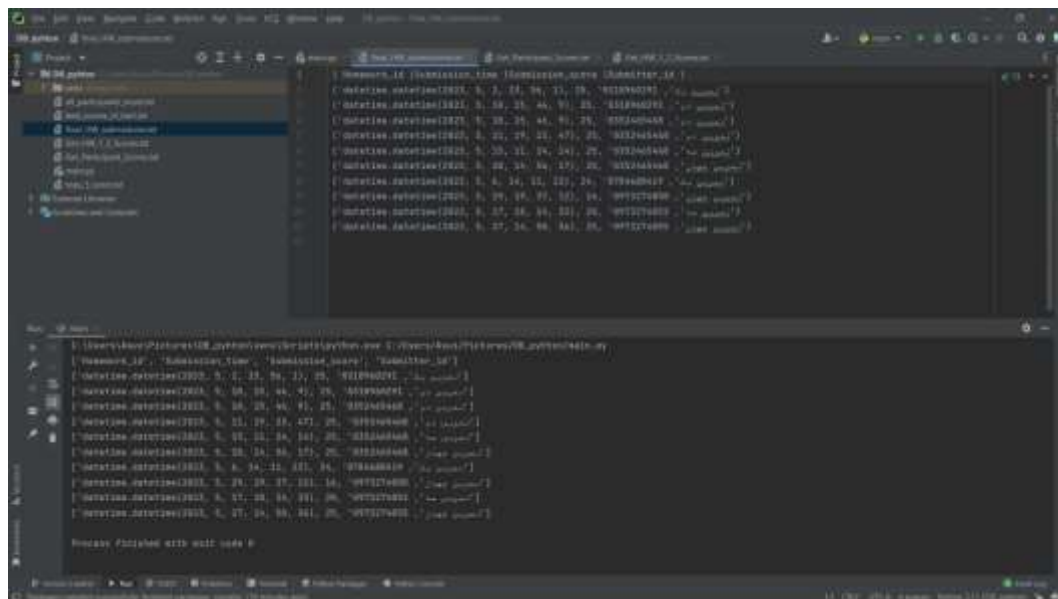
سابمیت‌های نهایی برای تمرین:



```
#include <iostream>
using namespace std;

int main() {
    int n, m;
    cin >> n >> m;
    int arr[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr[i][j];
        }
    }
    int max_val = -1;
    for (int i = 0; i < n; i++) {
        int max_row = -1;
        for (int j = 0; j < m; j++) {
            if (arr[i][j] > max_row) {
                max_row = arr[i][j];
            }
        }
        if (max_row > max_val) {
            max_val = max_row;
        }
    }
    cout << max_val << endl;
    return 0;
}
```

The screenshot shows a C++ IDE with a file named 'main.cpp'. The code defines a 2D array 'arr' of size 'n' by 'm'. It uses nested loops to read the values of the array. Then, it iterates through each row, finding the maximum value in that row. Finally, it compares the maximum of each row to find the overall maximum value in the entire array, which is then printed.

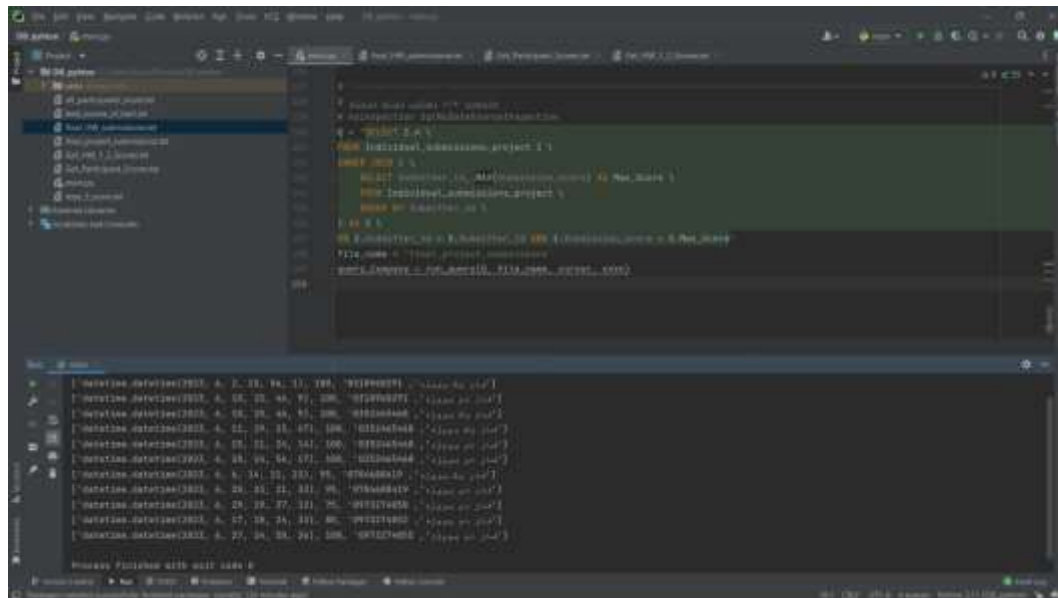


```
#include <iostream>
using namespace std;

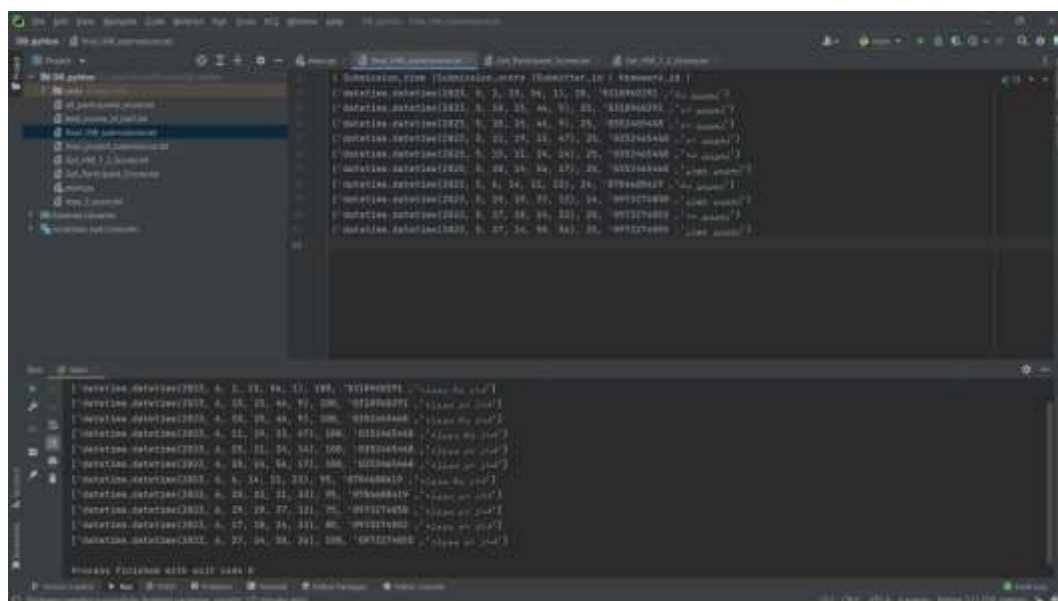
int main() {
    int n, m;
    cin >> n >> m;
    int arr[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> arr[i][j];
        }
    }
    int max_val = -1;
    for (int i = 0; i < n; i++) {
        int max_row = -1;
        for (int j = 0; j < m; j++) {
            if (arr[i][j] > max_row) {
                max_row = arr[i][j];
            }
        }
        if (max_row > max_val) {
            max_val = max_row;
        }
    }
    cout << max_val << endl;
    return 0;
}
```

This screenshot is identical to the one above, showing the same C++ code in the IDE. The code is a solution for finding the maximum value in a 2D array by iterating through each row and finding the maximum in that row, then comparing these to find the overall maximum.

سابمیت‌های نهایی برای پروژه:

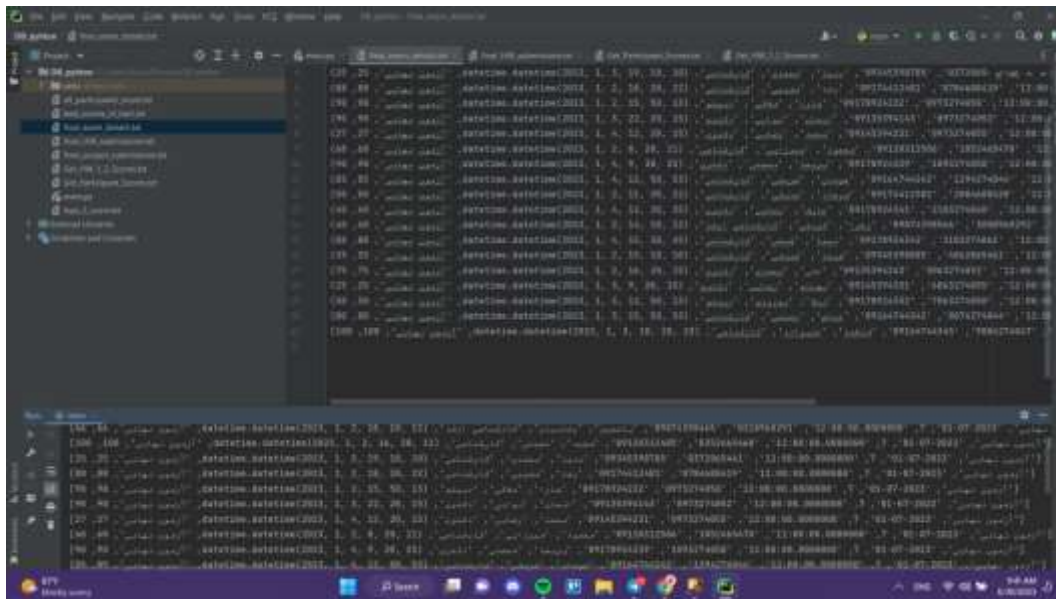
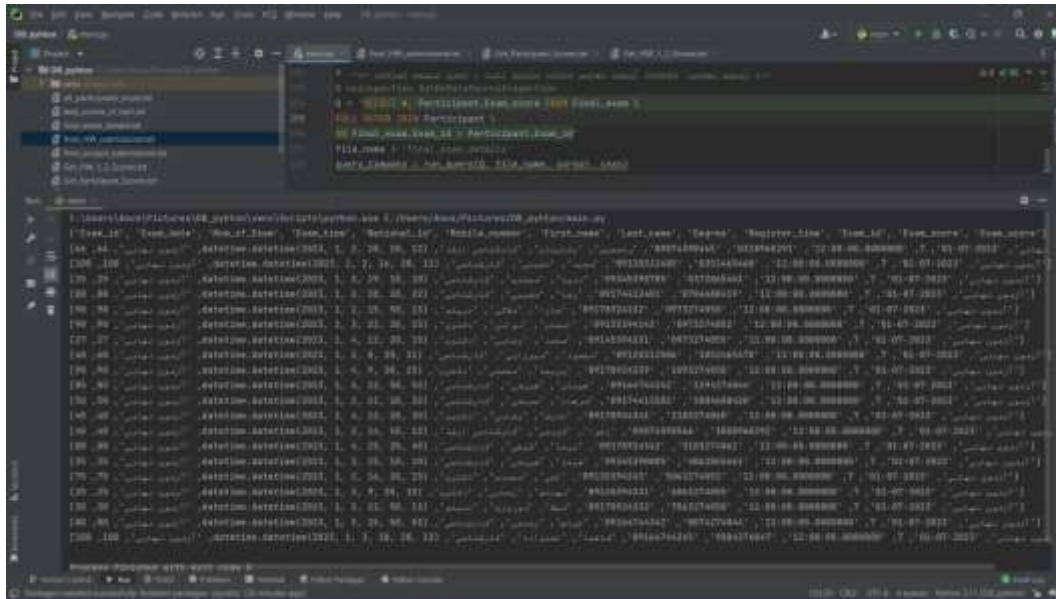


```
def submit(test_data):  
    # Create a RandomForestRegressor model  
    model = RandomForestRegressor()  
    # Train the model on training data  
    model.fit(train_data, train_labels)  
    # Predict the labels for the test data  
    predictions = model.predict(test_data)  
    # Return the predictions as a list  
    return predictions
```



```
def submit(test_data):  
    # Create a RandomForestRegressor model  
    model = RandomForestRegressor()  
    # Train the model on training data  
    model.fit(train_data, train_labels)  
    # Predict the labels for the test data  
    predictions = model.predict(test_data)  
    # Return the predictions as a list  
    return predictions
```

درج آزمون نهایی: مشخصات آزمون نهایی مانند تاریخ اجرا و نمره ورودی خواهند بود:



انتخاب ۵ نفر اول تمرین بر اساس نمره داده شده به آن‌ها:

The screenshot shows a Jupyter Notebook with a SQL query in the code cell. The query is as follows:

```

SELECT TOP 5 Participant_ID,
       First_Name, Last_Name, Participant_Score,
       Title_Name = "Top 5 Score"
FROM (
    SELECT Participant_ID,
           First_Name, Last_Name, Participant_Score
    FROM Participant_Score
    ORDER BY Participant_Score DESC
) AS RankedParticipants
ORDER BY Participant_ID;

```

The output cell shows the results of the query:

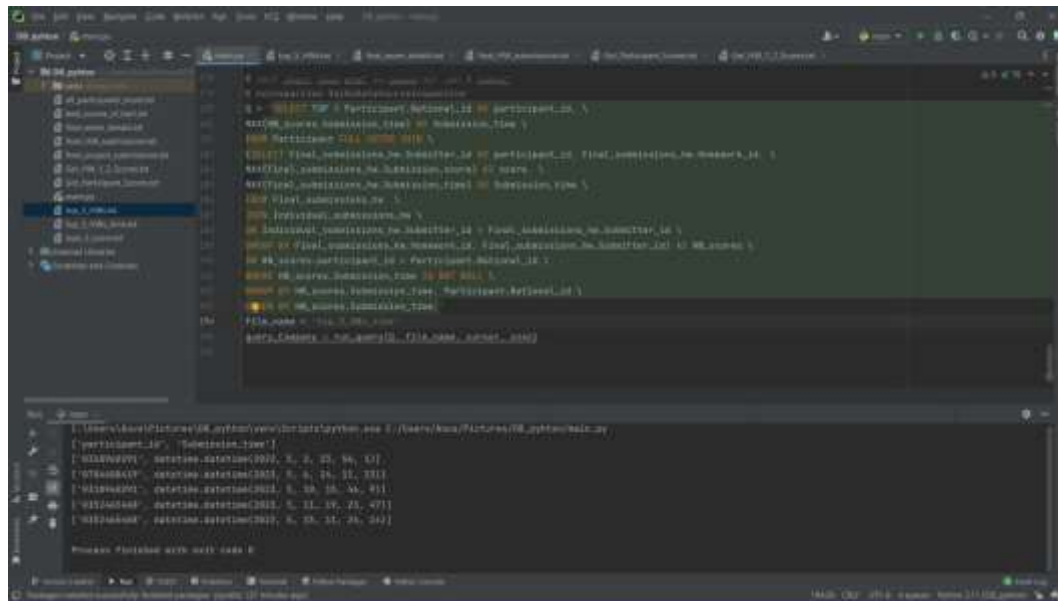
```

[[{"Participant_ID": 993216000, "First_Name": "John", "Last_Name": "Doe", "Participant_Score": 95},
 {"Participant_ID": 993216001, "First_Name": "Jane", "Last_Name": "Smith", "Participant_Score": 90},
 {"Participant_ID": 993216002, "First_Name": "Mike", "Last_Name": "Johnson", "Participant_Score": 85},
 {"Participant_ID": 993216003, "First_Name": "Emily", "Last_Name": "Brown", "Participant_Score": 80},
 {"Participant_ID": 993216004, "First_Name": "David", "Last_Name": "Wilson", "Participant_Score": 75}]]

```

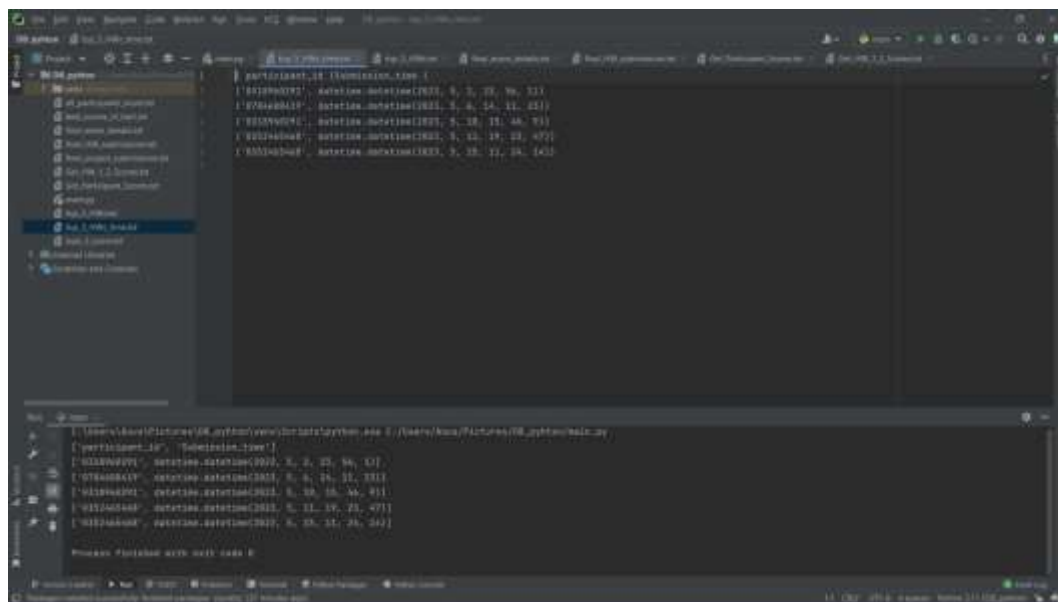
This screenshot is identical to the one above, showing the same SQL query and its output in a Jupyter Notebook environment.

انتخاب ۵ نفر اول تمرین بر اساس زمان ارسال آن‌ها:



```
def solve():
    n = int(input())
    for i in range(1, n+1):
        participant_id = input()
        submission_time = int(input())
        if submission_time < 0:
            continue
        if submission_time < min_time:
            min_time = submission_time
            min_participant_id = participant_id
    print(min_participant_id, min_time)
```

Process finished with exit code 0



```
def solve():
    n = int(input())
    for i in range(1, n+1):
        participant_id = input()
        submission_time = int(input())
        if submission_time < 0:
            continue
        if submission_time < min_time:
            min_time = submission_time
            min_participant_id = participant_id
    print(min_participant_id, min_time)
```

Process finished with exit code 0

انتخاب ۵ نفر برتر هر تمرین بر اساس نمره و زمان ارسالی آن‌ها:

[illegible][illegible]

مراجع

مراجع

- [1] Quera (Quera.org)



**Sharif University of Technology
Tehran Branch
Department of Computer Engineering**

**Proposal for Database Course Project
Quera Algorithm Bootcamp**

Title:

Quera Algorithm Bootcamp

Supervisor:

Mahdi Dadbakhsh

By:

Donya Navabi 99170597

Shayan Salehi 99105561

Iman Mohammadi 99102207

March 2023