

دانشگاه سهند

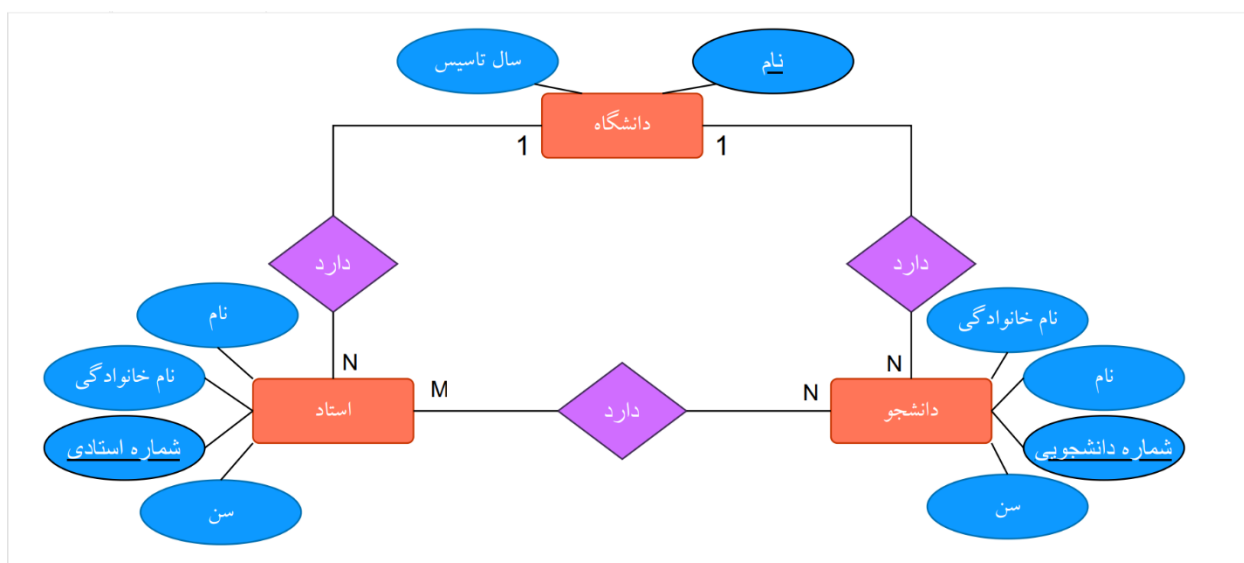
گزارش کار تمرین اول آزمایشگاه پایگاه داده

استاد: آرش یوسفی

گروه ۱

الف)

محیط عملیات دانشگاه را انتخاب کردیم، و سه موجودیت "دانشگاه"، "استاد" و "دانشجو" را برای آن در نظر گرفتیم. روابط بین موجودیت ها را مشخص کردیم؛ و برای هر موجودیت یک صفت شناسه در نظر گرفتیم.



ب)

در محیط psql، با استفاده از دستور "create database <Database Name>" یک دیتابیس جدید به نام university ایجاد کردیم. سپس با استفاده از دستور "\l" اطمینان حاصل کردیم که دیتابیس ایجاد شده است.

```
postgres=# create database university;
CREATE DATABASE
postgres=# \l
```

| Name | Owner | Encoding | Collate | List of databases Ctype | ICU Locale | Locale Provider | Access privileges |
|---------------|----------|----------|----------------------------|----------------------------|------------|-----------------|----------------------|
| d2 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | | libc | |
| postgres | postgres | UTF8 | English_United States.1252 | English_United States.1252 | | libc | |
| template0 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | | libc | =c/postgres |
| res | | | | | | | postgres |
| =CTc/postgres | | | | | | | |
| template1 | postgres | UTF8 | English_United States.1252 | English_United States.1252 | | libc | =c/postgres |
| res | | | | | | | postgres |
| =CTc/postgres | | | | | | | |
| university | postgres | UTF8 | English_United States.1252 | English_United States.1252 | | libc | |

(5 rows)

(ج)

با استفاده از دستور “\c university” به دیتابیس ساخته شده، متصل می‌شویم. سپس موجودیت‌ها را با استفاده از ساختن table در دیتابیس ایجاد و صفتهای هر کدام را تعریف می‌کنیم.

```
postgres=# \c university
You are now connected to database "university" as user "postgres".
university=# create table student (
university(# Fname varchar(15) not null,
university(# Lname varchar(20) not null,
university(# SID char(10) primary key,
university(# age smallint check(age > 0 and age < 120)
university(# );
CREATE TABLE
```

```
university=# create table Professor (
university(# Fname varchar(15) not null,
university(# Lname varchar(20) not null,
university(# PID char(10) primary key,
university(# age smallint check(age > 0 and age < 120)
university(# );
CREATE TABLE
university=# create table University (
university(# name varchar(20) primary key,
university(# Eyear int check(Eyear > 1200 and Eyear < 1402));
CREATE TABLE
university=# \d
                List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | professor      | table | postgres
 public | student        | table | postgres
 public | university     | table | postgres
(3 rows)
```

(د)

با دستور “insert into <TableName> values” رکوردهایی را به جداول ایجاد شده اضافه کردیم. در سه تصویر اول درج تک سطری، و در تصویر آخر درج چند سطری انجام شده است.

```
university=# insert into student values
university-# ('Alireza', 'Afshar', '9801226815', 23);
INSERT 0 1
```

```
university=# insert into University values
university-# ('Guilan',1355);
INSERT 0 1
```

```
university=# insert into professor values
university-# ('Farid','Feyzi','32',37);
INSERT 0 1
```

```
university=# insert into student values
university-# ('Parsa',
university(# 'Niroomand',
university(# '9801226853',
university(# 24);
INSERT 0 1
```

(ه)

در این قسمت، یک استاد را با سن بیش از ۱۲۰ سال تعریف کردیم؛ و مشاهده می‌کنیم که با خطا مواجه می‌شویم. زیرا در هنگام تعریف table استاد، سن را با شرط زیر ۱۲۰ سال تعریف کرده بودیم.

```
university=# insert into professor values
university-# ('Mohammad','Salehi','33',123);
ERROR:  new row for relation "professor" violates check constraint "professor_age_check"
DETAIL:  Failing row contains (Mohammad, Salehi, 33      , 123
).
university=# |
```

(ن)

از دستور “select * from <TableName>” برای نمایش تمامی رکوردهای ثبت شده در یک جدول استفاده می‌کنیم.

```
university=# select * from student
university=# ;
  fname |  lname  |   sid   | age
-----+-----+-----+----
Alireza | Afshar   | 9801226815 | 23
Parsa   | Niroomand | 9801226853 | 24
(2 rows)
```

(و)

در ابتدا دو رکورد دیگر به جدول student اضافه کردیم. سپس آنها را با استفاده از دستور ذکر شده در قسمت قبل، نمایش دادیم. در انتها شرطی را با استفاده از دستور select ایجاد کردیم که فقط رکوردهایی را که سن آنها بیشتر از ۲۳ است را نمایش دهد.

```
university=# select * from student ;
  fname |  lname  |   sid   | age
-----+-----+-----+----
Alireza | Afshar   | 9801226815 | 23
Parsa   | Niroomand | 9801226853 | 24
Soroosh | Gholizadeh | 9801226874 | 33
Sara    | Akhlaqi  | 9801226875 | 17
(4 rows)

university=# select * from student where age > 23 ;
  fname |  lname  |   sid   | age
-----+-----+-----+----
Parsa   | Niroomand | 9801226853 | 24
Soroosh | Gholizadeh | 9801226874 | 33
(2 rows)
```

ص، ض)

در اولین دستور با استفاده از تابع `char_length()` که در داکيومنت PostgreSQL پیدا کردیم، در جدول `student`، فقط رکوردهایی را که طول حروف اسم آنها بیشتر از ۴ است را نمایش بدهد.

در دستور دوم مشخص کردیم که رکوردی که حروف اسم آن دو 0 پشت سر هم دارد را نمایش دهد.

```
university=# select * from student where char_length(fname)>4;
  fname |  lname  |  sid   | age
-----+-----+-----+----
Alireza | Afshar  | 9801226815 | 23
Parsa   | Niroomand | 9801226853 | 24
Soroosh | Gholizadeh | 9801226874 | 33
(3 rows)
```

```
university=# select * from student where Fname like '%oo%';
  fname |  lname  |  sid   | age
-----+-----+-----+----
Soroosh | Gholizadeh | 9801226874 | 33
(1 row)
```

ط)

نخست تمامی رکورد های ثبت شده در جدول `student` را نمایش دادیم. سپس با دستور `update`، نام دانشجویی با شماره دانشجویی "۹۸۰۱۲۲۶۸۰۰۵۳" را تغییر دادیم.

پس از نمایش دوباره تمامی رکورد های ثبت شده، مشاهده می کنیم که نام دانشجوی مورد نظر تغییر داده شده است.

```
university=# select * from student ;
```

| fname | lname | sid | age |
|---------|------------|------------|-----|
| Alireza | Afshar | 9801226815 | 23 |
| Parsa | Niroomand | 9801226853 | 24 |
| Soroosh | Gholizadeh | 9801226874 | 33 |
| Sara | Akhlaqi | 9801226875 | 17 |

(4 rows)

```
university=# update student set fname='mohammad' where sid='9801226853';
UPDATE 1
university=# select * from student;
```

| fname | lname | sid | age |
|----------|------------|------------|-----|
| Alireza | Afshar | 9801226815 | 23 |
| Soroosh | Gholizadeh | 9801226874 | 33 |
| Sara | Akhlaqi | 9801226875 | 17 |
| mohammad | Niroomand | 9801226853 | 24 |

(4 rows)

(ظ)

با استفاده از دستور delete، از جدول student، رکوردی با نام “Alireza” را حذف کردیم. پس از نمایش تمامی رکوردها مشاهده می‌شود که رکورد مورد نظر حذف شده است.

```
university=# delete from student where fname='Alireza';
DELETE 1
university=# select * from student;
```

| fname | lname | sid | age |
|----------|------------|------------|-----|
| Soroosh | Gholizadeh | 9801226874 | 33 |
| Sara | Akhlaqi | 9801226875 | 17 |
| mohammad | Niroomand | 9801226853 | 24 |

(3 rows)

(ع)

در PostgreSQL از یک Tablespace برای نگاشت (map) یک نام منطقی به یک مکان فیزیکی روی دیسک استفاده می‌شود. در ساده‌ترین کلمه، می‌توانیم Tablespace را به عنوان مکانی روی دیسک در نظر بگیریم که تمامی object های پایگاه داده PostgreSQL در آن ذخیره می‌شوند. این object ها می‌توانند یک فهرست، یک جدول و غیره باشند.

PostgreSQL دارای دو Tablespace پیش فرض است:

❖ pg_defaulttablespace : برای ذخیره اطلاعات کاربر استفاده می‌شود.

❖ pg_globaltablespace : برای ذخیره داده های global استفاده می‌شود.

به صورت زیر نیز تعریف می‌شود:

```
CREATE TABLESPACE tablespace_name
```

```
OWNER user_name
```

```
LOCATION directory_path;
```