

جزوه کار علمی پایگاه داده برای استفاده از SQL

تهیه کنندگان:

سجاد مظاهری - مرتضی سرگلزائی جوان

Table

```
Create Table TableName ( parametr1 char(10) primary key not null,
                        parametr2 varchar(20),
                        parametr3 int,
                        ...
)
```

Exmample1:

ایجاد جدول اطلاعات دانشجو با فیلدهای شماره دانشجویی، نام، رشته و آدرس

```
Create Table Student( Sid varchar(10) primary key not null,
                      Sname varchar(20),
                      Major int,
                      Address text,
)
```

Exmample2:

ایجاد جدول اطلاعات ترم با فیلدهای شماره درس، فیلد خارجی شماره دانشجویی که درس را اخذ کرده، شماره ترم، نمره

```
Create Table Term( Cid varchar(10),
                  Sid varchar(10) foreign key not null references
                  Student(Sid),
                  TermNum varchar(10),
                  Grade int
)
```

Example3:

ایجاد جدول اطلاعات مولف با فیلدهای شماره شناسایی، تاریخ تولد، مدرک (شماره شناسایی کلید خارجی به جدول شخص می باشد و به صورت آنبشاری تغییرات را اعمال می کند.)

```
create table Author( Aid varchar(10) not null primary key,
                    Brithday datetime,
                    Madrak varchar(10),
                    foreign key (Aid) references Person ON DELETE CASCADE
)
```

Example4:

ایجاد جدول کتاب با فیلدهای شماره کتاب (فیلد خارجی)، تعداد، ناشر، کد و تاریخ انتشار

```
create table Book( Bid varchar(10) not null primary key,
                  Number int,
                  Publisher varchar(10),
                  ISBN varchar(20),
                  Publish_Date datetime,
                  foreign key (Bid) references Mohtava ON DELETE CASCADE
)
```

Example5:

ایجاد جدول اطلاعات روزنامه با فیلدهای شماره (فیلد خارجی)، تعداد، تاریخ انتشار

```
create table Newspaper( Nid varchar(10) not null,
                       Number varchar(10) not null,
                       Date datetime,
                       foreign key (Nid) references Mohtava ON DELETE CASCADE
)
```

Type

تعریف نوع داده ای جدید

Example1:

```
CREATE TYPE Sid FROM varchar(11) NOT NULL
CREATE TYPE sname FROM varchar(20)
CREATE TYPE grade FROM numeric(4,2)
CREATE TYPE address FROM text
CREATE TABLE student2(id sid,name sname,addr address)
```

نحوه استفاده از داده های تعریف شده

Example2:

```
create type StdID From char(4) Not Null
create type StfName From char(20) Not Null
create type Mark From Numeric(4,2) not Null
create type Mark2 From Numeric(4,2) Not Null
```

تعریف متغیر

```
declare @a Mark
declare @b Mark2
```

مقدار دهی متغیر

```
set @a=12.4
set @b=@c+3
```

نمایش متغیر

```
select @a
select @b
```

ساخت جدول

```
create table t1 (id StdID,name StdName, avg Mark)
```

With

With : يك جدول موقتي است که در Query مربوط به خودش استفاده مي شود.

```
Create With WithName [(Parametr1, Parametr2,...)] as
(
    < Select Expression >
)
```

Exmple1:

جدول موقتي زير نام و رشته دانشجوياني را که درس هایشان را پاس کرده اند، را شامل مي شود.

```
With StuPass (StuName,StuMajor) as
(
    select Sname,major
    From Term
    Where Grade>=10
)
```

Query زير با استفاده از جدول موقتي، نام دانشجويان IT که دروس خود را پاس کرده اند را نشان مي دهد.

```
With StuPass (StuName,StuMajor) as
(
    select Sname,major
    From Term
    Where Grade>=10
)
select sname
from StuPass
where major='IT'
```

Example2:

جدول موقت زير شامل اطلاعات دانشجويان مي باشد.

```
With Stu (StuName, Major) as
(
    select Sname, Major
    from person join student on person.pid=student.pid
)
```

حال با استفاده از اين جدول، اطلاعات دانشجويان کامپيوتر را انتخاب مي کنيم.

```
With Stu (StuName, Major) as
(
    select Sname, Major
    from person join student on person.pid=student.pid
)
select stuname
from stu
where major='Computer'
```

View

دیدگاه بصورت دائمی در پایگاه داده ایجاد می شود و تنها با حذف از پایگاه داده، دیگر وجود نخواهد داشت.

```
Create View ViewName as
(
    < Select Expression >
)
```

Example1:

دیدگاه زیر درس هایی را که هر دانشجو پاس کرده است را ایجاد می کند.

```
create view StuPass as
(
    select *
    From Term,student
    Where term.sid=student.sid and Grade>=10
)
```

Query زیر دانشجویانی که درس پایگاه داده را پاس کرده ان را نشان می دهد.

```
select StuPass.*
from StuPass join course on StuPass.cid=Course.cid
where Course.cname='Data Base'
```

Example2:

دیدگاه زیر شامل نام و رشته دانشجویان می باشد.

```
create view personMajor as
(
    select Sname,Major
    from person join student on person.pid=student.pid
)
```

Query زیر نام دانشجویان رشته کامپیوتر را نشان می دهد.

```
select sname
from personMajor
where major='Computer'
```

Example2:

تغییر دیدگاه دانشجو به استاد

```
alter view personMajor as
(
    select fname,cid
    from person join professor on person.pid=professor.pid
)
```

Query زیر دانشجویان نام اساتیدی که درس پایگاه داده را تدریس می کنند، نشان می دهد.

```
select *
from stu join course on personMajor.cid=course.cid
where course.cname='Data Base'
```

Procedure

```
Create Procedure ProName [@Parametr1 Type,@Parametr2 Type,... ] as
(
    < Select Expression >
)
```

Example1:

Procedure زیر اطلاعات دانشجویان را انتخاب می کند.

```
Create Procedure memberSelection as
(
    Select *
    From person join student on person.pid=student.pid
)
memberSelection //Call Procedure
```

Example2:

Procedure زیر اطلاعات دانشجویان رشته کامپیوتر را انتخاب می کند.

```
Create Procedure stuMajor @Major varchar(10) as
(
    Select *
    From memberSelection
    Where Major=@Major
)
stuMajor 'Computer' //Call Procedure
```

Example3:

تغییر Procedure

```
Alter Procedure memberSelection as
(
    Select *
    From person join professor on person.pid= professor.pid
)
memberSelection //Call Procedure
```

Function

```

Create Function FuncName [(@Parametr1 Type,@Parametr2 Type,... )]
Returns Table
as
Return
(
    < Select Expression >
)
FuncName (@Parametr1 Type,@Parametr2 Type,... ) //call Function

```

Example1:

ایجاد تابع اطلاعات دانشجو با فیلد رشته انتخابی

```

Create Function memberMajorSelection(@major varchar(10))
returns table
as
return
(
    Select student.*
    From person join student on person.pid=student.pid
    where major=@major
)
select * from memberMajorSelection('Computer')

```

Example2:

تغییر تابع

```

Alter Function memberMajorSelection(@cname varchar(10))
returns table
as
return
(
    Select professor.*
    From person join professor on person.pid=professor.pid
    Join course on professor.cid=course.cid
    where cname=@cname
)
select * from memberMajorSelection('C#')

```

Trigger

Structure:

```
CREATE TRIGGER triggerName
ON { table | view }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS {
    < SQL statement >
}
```

Example1: --Delete Student

```
create trigger upStu2 on t1 for Update
as
    select * from t1
```

Example2:

```
alter trigger upStu on t1 for Update
as
    update t1
    set avg=avg+1
    where id in (
        select id
        from inserted i)
    update t1
    set avg=avg-3
    where id in (
        select id
        from deleted i)
update t1
set avg=avg+1
where name='a'
```

Transaction

در تراکنش، در صورت بوجود آمدن خطا در هنگام انجام عملیات، عملیات انجام شده قبل از خطا را اعمال نمی کند.

```
BEGIN { TRAN | TRANSACTION } [ transaction_name ]
    < Expression >
Commit {TRAN | TRANSACTION } [ transaction_name ]
```

Example1:

در مثال زیر به دلیل اینکه تعداد کارکترهای 31010 بیش از حد مجاز می باشد رخ خواهد داد بنابراین عملیات قبلی اعمال نخواهد شد.

```
begin try
    begin transaction
        insert into t1 values ('1011', 'a', 10)
        select max(avg)
        from t1

        insert into t1 values ('31010', 'a', 15)

        select min(avg)
        from t1
    commit transaction
end try
begin catch
    rollback
end catch
```

MISC

برخی دستورات و توابع کاربردی دیگر

```
PRINT 'message' --print message
RAISERROR --print error
DROP TABLE student - delete Table
DECLARE @variable [AS] data_type --definition variable
SET @local_variable = data -- set data
```

Example:

تعریف و مقدار دهی متغیر

```
declare @c int
set @C=0
```

دستورات شرطی

```
IF @c > 0
    print 'The number is greater than zero'
else
    IF @c < 0
        print 'The number is less than zero'
else
```

نمایش پیغام

```
PRINT 'message'
```

نمایش تاریخ فعلی سیستم

```
PRINT GetDate()
```

نمایش خطا

```
RAISERROR (N'This is message %s %d.',10,1,N'number',5)
```

نمونه سئوالات SQL – سری اول

طراحی سیستم آموزشی زیر را در نظر بگیرید. (توضیح: برخی سئوالات به چند روش حل شده اند)

Student(Sid, Sname , City, Major)
 Course(Cid, Cname, Credit, Major, Precid)
 Term(Sid , Cid , Pid , Grade , TermNo , Year)
 Professor (Pid , Pname , Degree , Major)

-----1-----
 ۱- اطلاعات دانشجویان رشته کامپیوتر که فارق التحصیل هستند عبارتی کلیه دروس مربوطه به گروه خود را پاس کرده اند.

```
with CompStu85(sid,cid,grade) as
(
    select term.sid, term.cid, grade
    from student join term
    on student.sid = term.sid
    where major='IT' and year=85 and termno=1 and grade>=10
)

select *
from student join
(
    select sid from compstu85
    except
    select sid from
    (
        select CompStu85.sid, ITCourse.cid
        from CompStu85, (Select cid from Course where major='IT') as ITCourse
        except
        select sid,cid
        from CompStu85
    ) as Temp
) as FinalStu
on student.sid = FinalStu.sid
```

-----1(2)-----

```
with std(sid,cnt) as
(
    select student.sid, count(cid) as cnt
    from student join term on student.sid = term.sid
    where major = 'IT' and grade>=10
    group by student.sid
)
select *
from std
where cnt in (select count(cid)
              from course
              where major='IT')
```

-----1(3)-----

```
select student.sname
from student join term on student.sid = term.sid
where major = 'IT' and grade>=10
group by student.sname
having count(cid) in (select count(cid)
                    from course
                    where major='IT')
```

```

--1(4)-----
select distinct student.*
from student join term
on student.sid = term.sid
where major = 'IT' and grade>=10 and not exists (
    select course.cid from course where course.major='IT'
    except
    select distinct term.cid from term where term.sid=student.sid)

```

2- تعداد دانشجویان مشروط هر رشته در ترم اول سال ۸۵ ؟

```

with Averages (sid,major,average) as
(
    select sid, major, sum(credit * grade)/sum(credit) as average
    from term join course
    on term.cid = course.cid
    group by sid, major
)
select major,count(average) as count
from Averages
where average<13
group by major

```

3- اطلاعات دانشجویان شاگرد اول هر رشته

```

with Averages (sid,major,average) as
(
    select student.sid, student.major, sum(grade*credit)/sum(credit)
as average
    from Term join Course on Term.cid=course.cid
    join student on student.sid = term.sid
    group by student.sid,student.major
)

select student.sid , student.sname, student.major, average
from student join Averages
on student.sid=Averages.sid, (select major,max(average) as maxAverage
                             from Averages
                             group by major) as MaxAverages
where student.Major = MaxAverages.Major and
Averages.average>=MaxAverages.maxAverage
order by maxAverage desc

```

```

--3(2)-----
with Averages (sid,major,average) as
(
    select student.sid, student.major, sum(grade*credit)/sum(credit)
as average
    from Term join Course on Term.cid=course.cid
    join student on student.sid = term.sid
    group by student.sid,student.major
)
select Averages.*
from Averages join (select major,max(average) as maxAverage
                   from Averages
                   group by major) as MaxAverages
on Averages.major = MaxAverages.major
where Averages.average>=MaxAverages.maxAverage
order by maxAverage desc

```

--4-----
 ۴- اطلاعات اساتیدی که تنها در همان گروه آموزشی تدریس می کنند و در سایر گروه ها تدریس ندارند؟

```
select *
from professor
except
select distinct Professor.PID, Professor.Pname, Degree, Professor.Major
from Professor join Term on Professor.PID = Term.PID
join course on course.cid = term.cid
where Professor.Major != course.Major
```

--4(2)-----
 select *
 from professor
 where major = all (
 select major
 from term join course
 on term.cid = course.cid
 where professor.pid = term.pid
 group by major
)

--5-----
 ۵- اطلاعات دانشجویان رشته کامپیوتر که در ترم اول سال ۸۵ (ترم جاری) درس پایگاه داده را بدون پیشنیاز آن اخذ کرده اند.

```
with PreValid(sid) as
(
    select term.sid
    from student join term
    on student.sid = term.sid
    where major='IT' and grade>=10 and term.cid in (select precid from course
where cname='DS')
)
select *
from student join
(
    select sid
    from term join course
    on term.cid = course.cid
    where course.cname='DS'
    except
    select *
    from PreValid
) as PreDump
on student.sid = PreDump.sid
```

نمونه سوال - سری ۲

B - به فرض اینکه رابطه زیر را برای ذخیره سازی اطلاعات کارمندان و مدیران سازمان در نظر بگیریم.
Employee(Eid , Name , Dept , MngID)

Eid Char(5): شماره کارمند	Name char(10): نام کارمند	Dept char(10): دپارتمان	MngId char(5): شماره مدیر مربوطه
---------------------------	---------------------------	-------------------------	----------------------------------

به عنوان مثال :

Eid	Name	dept	MngId
E1	Ali	Null	Null
E2	Reza	product	E1
E3	Ahmad	research	E1
E4	Hassan	Machin1	E2
E5	Majid	Machin2	E2

با استفاده از دستورات sql پرس و جویی Query بنویسید که کلیه کارمندانی که تحت مدیریت کارمندی بنام Reza و شماره E2 هستند را مشخص نماید این کارمندان شامل کارمندان رده های پایین تر نیز می شود؟
بعنوان مثال کارمندان Reza شامل E4, E5 می باشند.

```
-- answer a:
--alter Table Quize2b add CONSTRAINT [aa] foreign key ([MngID]) references
Quize2b ([Eid])

create table temp1(id char(5))
create table temp2(id char(5))
create table temp3(id char(5))

delete temp1
delete temp2
delete temp3

insert into temp1 select eid from Quize2b where eid='E2'
while exists (Select * from temp1)
begin
    insert into temp2
        select eid from Quize2b where MngID in (select id
from temp1)
    insert into temp3
        select * from temp2
    delete temp1
    insert into temp1
        select * from temp2
    delete temp2
end

select * from temp3

drop table temp1
drop table temp2
drop table temp3
```

نحوه اتصال به پایگاه داده در C#:

```
String strcon = "Data Source=Server_name;Initial Catalog=database_name;Integrated Security=True;"
private SqlConnection scon;

scon = new SqlConnection(strcon);
scon.Open();
```

مثال ۱- اجرای دستورات SQL:

```
String strcon = "Data Source=Server_name;Initial Catalog=database_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "SELECT Field1 FROM Table";
SqlDataReader reader= scom.ExecuteReader();
While(reader.Read())
{
    MessageBox.Show(reader.GetValue(0).ToString());
}
```

مثال ۲- اجرای دستورات SQL:

```
String strcon = "Data Source=Server_name;Initial Catalog=database_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "Delete FROM Table WHERE Field1='value'";
scom.ExecuteNonQuery();
```

مثال ۳- اجرای دستورات SQL:

```
String strcon = "Data Source=Server_name;Initial Catalog=database_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "INSERT INTO Table VALUES('value')";
scom.ExecuteNonQuery();
```

مثال ۴- اجرای دستورات SQL:

```
String strcon = "Data Source=Server_name;Initial Catalog=datebase_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "UPDATE Table SET Field1=value1+2";
scom.ExecuteNonQuery();
```

مثال ۵- ارسال پارامتر:

```
String strcon = "Data Source=Server_name;Initial Catalog=datebase_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "SELECT Field1 FROM Table1 WHERE ID='"+idTextBox.Text+"'";
scom.ExecuteNonQuery();
```

مثال ۶- ارسال پارامتر:

```
String strcon = "Data Source=Server_name;Initial Catalog=datebase_name;Integrated Security=True;"
private SqlConnection scon;
private SqlCommand scom;

scon = new SqlConnection(strcon);
scon.Open();

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.Text;
scom.CommandText = "SELECT Field1 FROM Table1 WHERE ID=@id";
scom.Parameters.Add("@id", SqlDbType.VarChar, 10).Value = idTextBox.Text;
scom.ExecuteNonQuery();
```

مثال ۷- اجرای رویه:

```
...

scom = new SqlCommand();
scom.Connection = scon;
scom.CommandType = CommandType.StoredProcedure;
scom.CommandText = "MyProc";
scom.Parameters.Add("@id", SqlDbType.VarChar, 10).Value = idTextBox.Text;
scom.Parameters.Add("@name", SqlDbType.VarChar, 20).Value = nameTextBox.Text;
... // ارسال سایر پارامترهای رویه
```

مثال ۸- نمایش اطلاعات در DataGridView:

```
SqlConnection scon;  
SqlCommand scom;  
SqlDataAdapter sda;  
  
scon = new SqlConnection(strconnection);  
scon.Open();  
scom = new SqlCommand();  
scom.Connection = scon;  
sda = new SqlDataAdapter(scom);  
scom.CommandType = CommandType.Text;  
  
DataTable dt = new DataTable();  
sda.Fill(dt);  
dataGridView1.DataSource = dt;
```

یک کلاس آماده جهت ایجاد سریع اتصال و افزایش سرعت کار با پایگاه داده

```
public class Connection : IDisposable
{
    public Connection()
    {
        scon = new SqlConnection(strcon);
        scon.Open();
        sconnected = true;
        econnected = false;
        scom = new SqlCommand();
        scom.Connection = scon;
        sda = new SqlDataAdapter(scom);
        scom.CommandType = CommandType.Text;
    }
    private static string strcon = "";
    public static string ConnectionString
    {
        set { strcon = value; }
        get { return strcon; }
    }
    public void Dispose()
    {
        Dispose(true);
    }
    private void Dispose(bool disposing)
    {
        scon.Close();
    }
    public SqlDataAdapter Adapter
    {
        get { return sda; }
        set { sda = value; }
    }
    public SqlCommand Command
    {
        set { scom = value; }
        get { return scom; }
    }
    public void FillDataGrid(DataGridView dgv)
    {
        DataTable dt = new DataTable();
        this.sda.Fill(dt);
        dgv.DataSource = dt;
    }
    private SqlConnection scon;
    private SqlCommand scom;
    private SqlDataAdapter sda;
}
```

نحوه استفاده:

Connection String را فقط یکبار و آنها در ابتدای برنامه بصورت زیر مقدار دهی کنید:

```
Connection.ConnectionString= "Data Source=Server_name;Initial Catalog=datebase_name;Integrated Security=True;"
```

سپس برای اتصال به پایگاه و اجرای دستورات فقط کافیهست بلاک زیر را ایجاد کنید:

```
using (Connection MyCon = new Connection())
{
    // دستورات در این قسمت نوشته میشود
}
```

مثال (١)

```
using (Connection MyCon = new Connection())
{
    MyCon.Command.CommandText= "Select Sid from Student";
    MyCon.FillDataGrid(dataGridView1);
}
```

مثال (٢)

```
using (Connection MyCon = new Connection())
{
    MyCon.Command.CommandText= "Select Sid from Student";
    SqlDataReader reader = MyCon.Command.ExecuteReader();
    while (reader.Read())
    {
        studentIdComboBox.Items.Add(reader.GetValue(0));
    }
    reader.Close();
}
```

مثال (٣)

```
using (Connection MyCon = new Connection())
{
    MyCon.Command.CommandType = CommandType.StoredProcedure;
    MyCon.Command.CommandText = "MyProc";
    scom.Parameters.Add("@id", SqlDbType.VarChar, 10).Value = idTextBox.Text;
    MyCon.FillDataGrid(dataGridView1);
}
```