



آزمایشگاه پایگاه داده

مدرس: مزیانیان

پایگاه داده

- ◆ مجموعه ای از داده های ذخیره شده به صورت مجتمع و مبتنی بر یک ساختار
- ◆ تعریف شده به صورت صوری
- ◆ تحت کنترل متمرکز
- ◆ مورد استفاده یک یا چند کاربر به طور اشتراکی و همزمان

سیستم مدیریت پایگاه داده (DBMS)

- ♦ یک مجموعه برنامه کامپیوتری برای مدیریت پایگاه داده و انجام عملیات درخواستی توسط کاربران متعدد بر روی داده ها
- ♦ فراهم کردن امکان کنترل برای دسترسی به داده، اجرای تمامیت داده ها، مدیریت کنترل همزمانی و بازیابی پایگاه داده پس از شکست و بازگرداندن آن از فایل های پشتیبان و همچنین حفظ امنیت پایگاه داده
- ♦ DBMS های معروف: Oracle، Microsoft SQL Server، PostgreSQL و MySQL

ساختارهای داده ای

◆ سلسله مراتبی

- ارتباطات بین داده ها به شکل والد و فرزندی
- برای مدلسازی ارتباطات یک به چند یک سویه

◆ شبکه ای

- رکوردها حاوی فیلدها و مجموعه ها رابطه های یک به چند بین رکوردها
- هر فرزند میتواند بیش از یک والد داشته باشد
- برای مدلسازی یک به چند دوسویه

◆ رابطه ای

- شامل رابطه ها، صفات خاصه و دامنه ها
- یک رابطه در قالب جدولی از سطر و ستون ها، هر ستون یک صفت خاصه و مجموعه مقادیر مجاز صفات خاصه دامنه ها هستند
- بر مبنای جبر رابطه ای

Basic Concepts

- ◆ Table
- ◆ Record (Tuple)
- ◆ Add , Remove , Modify , Search and Join tables in a database

سرفصل

- ◆ داندود و نصب MySQL
- ◆ آشنایی با محیط MySQL Workbench
- ◆ ایجاد و حذف پایگاه داده
- ◆ ایجاد و حذف جدول
- ◆ ویرایش ساختار جدول
- ◆ درج و حذف رکورد در جدول
- ◆ ویرایش رکورد در جدول
- ◆ ایجاد Query
- ◆ دستور Select در SQL
- ◆ توابع در SQL

دستورات SQL

♦ ایجاد یک پایگاه داده

Create database نام پایگاه داده

♦ حذف یک پایگاه داده

Drop database نام پایگاه داده

♦ ساخت جدول

Create table (نام جدول

نام ستون ۱ (اندازه) نوع

Null/Not null,

.

.

.

Primary key (نام ستون),

)

مثال

♦ جدول Student که به ترتیب شامل شماره دانشجویی، نام دانشجو، فامیل دانشجو، معدل و محل زندگی اوست.

```
CREATE TABLE `lab`.`Student` (  
  `Sid` INT NOT NULL,  
  `Sname` VARCHAR(45) NOT NULL,  
  `Sfamily` VARCHAR(45) NOT NULL,  
  `Address` VARCHAR(100) NULL,  
  PRIMARY KEY (`Sid`));
```


♦ جداول درس و استاد را با ستون های زیر ایجاد کنید

Course = (cid, cname, ctype, unit)

Prof = (pid, pname, esp, degree, city)

تعریف کلید خارجی

نام ستون
cascade
on delete cascade
not null references
on update (نام ستون) نام جدول

♦ اضافه کردن جدول grade

```
CREATE TABLE `lab`.`grade` (  
  `Sid` INT NOT NULL,  
  `Cid` INT NOT NULL,  
  `Pid` INT NOT NULL,  
  `Grade` DOUBLE NULL,  
  PRIMARY KEY (`Sid`, `Cid`, `Pid`),
```

```
  CONSTRAINT `St`
```

```
    FOREIGN KEY (`Sid`) REFERENCES `lab`.`student` (`Sid`)  
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
  CONSTRAINT `C`
```

```
    FOREIGN KEY (`Cid`) REFERENCES `lab`.`course` (`Cid`)  
    ON DELETE CASCADE ON UPDATE CASCADE,
```

```
  CONSTRAINT `p`
```

```
    FOREIGN KEY (`Pid`) REFERENCES `lab`.`prof` (`Pid`)  
    ON DELETE CASCADE ON UPDATE CASCADE);
```

تغییر در ساختار جدول

– اضافه کردن یک ستون

نام جدول `Alter table`

`Add Column` نام ستون (اندازه) نوع `null;`

`Alter table student`

`Add fathername char (20) null;`

تغییر در ساختار جدول

– حذف یک ستون

نام جدول `Alter table`

نام ستون ۲, نام ستون ۱ `Drop column` ; ...

– تغییر مشخصات یک ستون

نام جدول `Alter table`

نوع جدید نام ستون نام ستون `Change column` ;

– حذف کلی یک جدول

نام جدول `Drop table` ;

قيدها (Constraints)

NOT Null Constraint ◆

Unique Constraint ◆

Default Constraint ◆

Primary Key Constraint ◆

Foreign Key Constraint ◆

Default Constraint

```
CREATE TABLE Persons(  
P_Id int NOT NULL UNIQUE,  
LastName varchar(255) NOT NULL,  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255) DEFAULT 'Shahrood'  
)
```

Default Constraint

```
ALTER TABLE Persons  
CHANGE COLUMN City City varchar(255) DEFAULT  
'Shahrood';
```

```
ALTER TABLE Persons  
CHANGE COLUMN City City varchar(255);
```


Unique Constraint

```
ALTER TABLE Persons  
ADD UNIQUE INDEX `persons_lastname_UNIQUE`;
```

```
ALTER TABLE Persons  
Drop Index `persons_lastname_UNIQUE`;
```

وارد کردن داده به جدول

Insert into نام جدول
(لیست ستون ها)

Values (لیست مقادیر)

```
insert into student  
(sid,sname,sfamily,savg,city)  
values  
(1000,'ali','karimi',14.28,'ahwaz')
```

وارد کردن داده به جدول

نام جدول Insert into

لیست مقادیر Values

```
insert into student  
values  
(1002,'maryam','mosavi',17.45,'karaj')
```

```
insert into student  
values  
(1003,'sina','mahdavi',11.33,null)
```

حذف داده از جدول

Delete

From نام جدول

Where شرط

```
delete  
from student  
where sid=1001
```


پرس و جوی ساخت یافته

بازیابی داده ها از جدول

Select ... , نام ستون ۲ , نام ستون ۱

[From نام جدول(ها)]

[Where شرط]

[Group by نام ستون]

[Having شرط]

[Order by نام ستون]

Books


BID	BName	Author	Publisher	Price	Nums
1	Internet	Ulman	Berekely	18000	2
2	Access	Elmasri	Wrox	20000	3
3	Database	C.J.Date	MIT	25000	1
4	Network	Elmasri	MIT	18000	2
5	Microsoft Access	C.J.Date	Press	10000	10
6	UML	Elmasri	Berekely	5000	5

Members

MID	MFName	MLName	MFaName	BirthDate
1	Hamidreza	Niroomand	Mostafa	1986
2	Yahya	Ghanbary	Hossein	1980
3	Ali	Radmard	Hojjat	1970
4	Ali	Akbary	Reza	2000
5	Maryam	Rezaee	Mahmood	2003

Lending

LID	BID	MID	L_Date	R_Date
1	2	3	2010/3/3	2010/3/10
2	1	4	2010/3/10	
3	5	5	2010/3/12	2010/3/17
4	3	3	2010/3/3	
5	2	1	2010/3/20	
6	5	1	2010/3/20	

پرس و جوی عمومی 

```
SELECT *  
FROM books;
```

```
SELECT members.mfname, members.mLname  
FROM members;
```

عملگرهای قابل استفاده در قسمت Where ♦

– عملگرهای مقایسه ای: <, >, <>, =<, =>, !=, =, !<, !>

– عملگرهای منطقی: AND , OR , NOT

SELECT * FROM members **WHERE** members.mfname='ali' AND members.BirthDate<1980;

– عملگر Between برای انتخاب بین چند گزینه

SELECT * FROM books **WHERE** books.price BETWEEN 20000 AND 30000;

– عملگر IN برای تعیین شرط با محدوده ای از مقادیر

SELECT * FROM books **WHERE** books.publisher
IN('wrox','harvard','mit','lynda');

Order by

♦ مرتب سازی خروجی نسبت به ستون ذکر شده

```
SELECT bname FROM books  
ORDER BY books.bname;
```

```
SELECT * FROM members  
ORDER BY members.BirthDate desc;
```

Group by

♦ دسته بندی خروجی بر اساس یک ستون

```
SELECT bid FROM lending GROUP BY bid;
```

♦ همراه با این عملگر می توان از توابع جمعی استفاده کرد

AVG –

Min , Max –

Count –

Sum –



```
SELECT author,AVG(price) AS AVG FROM books
```

```
GROUP BY author;
```

```
SELECT author,MIN(price) AS Min FROM books
```

```
GROUP BY author;
```

```
SELECT COUNT(*) AS Cnt FROM books;
```

```
SELECT SUM(nums)AS Sum FROM books
```

```
WHERE author='elmasri';
```

```
SELECT SUM(price*nums) FROM books;
```


Having

♦ حتما با Group by می آید

♦ پس از گروه بندی روی گروه ها اعمال می شود

```
SELECT author FROM books GROUP BY author HAVING  
SUM(nums)>10;
```

Distinct , All

◆ گزاره ALL به طور پیش فرض برای Select در نظر گرفته می شود

◆ گزاره Distinct مقادیر تکراری را نمایش نمی دهد

```
SELECT DISTINCT author FROM books;
```

Like

♦ توسط این عملگر یک الگو یا شکل کلی برای ارزیابی مقادیری که میخواهیم، تعریف میکنیم که در صورت تطابق نمایش داده شوند

```
SELECT * FROM books where Publisher like '%re%'
```

JOIN

◆ **Cross Join** : ایجاد لیستی از تمام سطرها که هر سطر از جدول اول با هر سطر از جدول دوم ارتباط یک به یک دارد

◆ **Inner Join** : دو جدول طوری ادغام می شوند که مقادیر غیر Null مرتبط را نمایش می دهند

◆ **Outer Join** : مقادیر Null مرتبط را هم برمی گرداند

– **Left Outer Join** : مقادیری که در جدول سمت چپ موجود و در جدول سمت راست Null است را نیز برمی گرداند

– **Right Outer Join** : مقادیری که در جدول سمت راست موجود و در جدول سمت چپ Null است را نیز برمی گرداند

– **Full Outer Join** : همه سطرها را برمی گرداند و مقادیر در صورت عدم وجود نظیر Null می شوند


```
SELECT * FROM books Cross Join Lending;
```

Results


Messages

BID	Bname	Author	Publisher	Price	Nums	LID	BID	MID	L_Date	R_Date
1	Internet	Ulman	Berekely	18000	2	1	2	3	2010-03-03	2010-03-10
2	Access	Elmasri	Wrox	20000	3	1	2	3	2010-03-03	2010-03-10
3	Database	C.J.Date	MIT	25000	1	1	2	3	2010-03-03	2010-03-10
4	Network	Elmasri	MIT	18000	2	1	2	3	2010-03-03	2010-03-10
5	Microsoft Access	C.J.Date	Press	10000	10	1	2	3	2010-03-03	2010-03-10
6	UML	Elmasri	Berekely	5000	5	1	2	3	2010-03-03	2010-03-10
1	Internet	Ulman	Berekely	18000	2	2	1	4	2010-03-10	NULL
2	Access	Elmasri	Wrox	20000	3	2	1	4	2010-03-10	NULL
3	Database	C.J.Date	MIT	25000	1	2	1	4	2010-03-10	NULL
4	Network	Elmasri	MIT	18000	2	2	1	4	2010-03-10	NULL
5	Microsoft Access	C.J.Date	Press	10000	10	2	1	4	2010-03-10	NULL
6	UML	Elmasri	Berekely	5000	5	2	1	4	2010-03-10	NULL
1	Internet	Ulman	Berekely	18000	2	3	5	5	2010-03-12	2010-03-17
2	Access	Elmasri	Wrox	20000	3	3	5	5	2010-03-12	2010-03-17
3	Database	C.J.Date	MIT	25000	1	3	5	5	2010-03-12	2010-03-17
4	Network	Elmasri	MIT	18000	2	3	5	5	2010-03-12	2010-03-17
5	Microsoft Access	C.J.Date	Press	10000	10	3	5	5	2010-03-12	2010-03-17
6	UML	Elmasri	Berekely	5000	5	3	5	5	2010-03-12	2010-03-17
1	Internet	Ulman	Berekely	18000	2	4	3	3	2010-03-03	NULL
2	Access	Elmasri	Wrox	20000	3	4	3	3	2010-03-03	NULL
3	Database	C.J.Date	MIT	25000	1	4	3	3	2010-03-03	NULL
4	Network	Elmasri	MIT	18000	2	4	3	3	2010-03-03	NULL
5	Microsoft Access	C.J.Date	Press	10000	10	4	3	3	2010-03-03	NULL


```
SELECT * FROM Books inner Join Lending ON Books.BID=Lending.BID;
```

III

Results

 Messages

	BID	Bname	Author	Publisher	Price	Nums	LID	BID	MID	L_Date	R_Date
	2	Access	Elmasri	Wrox	20000	3	1	2	3	2010-03-03	2010-03-10
	1	Internet	Ulman	Berekely	18000	2	2	1	4	2010-03-10	NULL
	5	Microsoft Access	C.J.Date	Press	10000	10	3	5	5	2010-03-12	2010-03-17
	3	Database	C.J.Date	MIT	25000	1	4	3	3	2010-03-03	NULL
	2	Access	Elmasri	Wrox	20000	3	5	2	1	2010-03-20	NULL
	5	Microsoft Access	C.J.Date	Press	10000	10	6	5	1	2010-03-20	NULL

```
SELECT * FROM Books Left Outer Join Lending ON Books.BID=Lending.BID;
```

III

Results



Messages

BID	Bname	Author	Publisher	Price	Nums	LID	BID	MID	L_Date	R_Date
1	Internet	Ulman	Berekely	18000	2	2	1	4	2010-03-10	NULL
2	Access	Elmasri	Wrox	20000	3	1	2	3	2010-03-03	2010-03-10
2	Access	Elmasri	Wrox	20000	3	5	2	1	2010-03-20	NULL
3	Database	C.J.Date	MIT	25000	1	4	3	3	2010-03-03	NULL
4	Network	Elmasri	MIT	18000	2	NULL	NULL	NULL	NULL	NULL
5	Microsoft Access	C.J.Date	Press	10000	10	3	5	5	2010-03-12	2010-03-17
5	Microsoft Access	C.J.Date	Press	10000	10	6	5	1	2010-03-20	NULL
6	UML	Elmasri	Berekely	5000	5	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM Books Right Outer Join Lending ON Books.BID=Lending.BID;
```

III

Results



Messages

	BID	Bname	Author	Publisher	Price	Nums	LID	BID	MID	L_Date	R_Date
	2	Access	Elmasri	Wrox	20000	3	1	2	3	2010-03-03	2010-03-10
	1	Internet	Ulman	Berekely	18000	2	2	1	4	2010-03-10	NULL
	5	Microsoft Access	C.J.Date	Press	10000	10	3	5	5	2010-03-12	2010-03-17
	3	Database	C.J.Date	MIT	25000	1	4	3	3	2010-03-03	NULL
	2	Access	Elmasri	Wrox	20000	3	5	2	1	2010-03-20	NULL
	5	Microsoft Access	C.J.Date	Press	10000	10	6	5	1	2010-03-20	NULL

```
SELECT * FROM Books Full Outer Join Lending ON Books.BID=Lending.BID;
```

III

Results



Messages

BID	Bname	Author	Publisher	Price	Nums	LID	BID	MID	L_Date	R_Date
1	Internet	Ulman	Berekely	18000	2	2	1	4	2010-03-10	NULL
2	Access	Elmasri	Wrox	20000	3	1	2	3	2010-03-03	2010-03-10
2	Access	Elmasri	Wrox	20000	3	5	2	1	2010-03-20	NULL
3	Database	C.J.Date	MIT	25000	1	4	3	3	2010-03-03	NULL
4	Network	Elmasri	MIT	18000	2	NULL	NULL	NULL	NULL	NULL
5	Microsoft Access	C.J.Date	Press	10000	10	3	5	5	2010-03-12	2010-03-17
5	Microsoft Access	C.J.Date	Press	10000	10	6	5	1	2010-03-20	NULL
6	UML	Elmasri	Berekely	5000	5	NULL	NULL	NULL	NULL	NULL

دید (View)

- ♦ دسترسی گروه های خاصی از کاربران به ستون های خاصی از چند جدول
- ♦ یک جدول مجازی
- ♦ تمامی داده ها در جداول اصلی
- ♦ نتیجه یک Select است
- ♦ برخورد SQL با View همانند یک جدول حقیقی

CREATE VIEW AS نام دید

SELECT نام ستون(ها)

FROM نام جدول

WHERE شرط


```
CREATE VIEW view_1 AS  
SELECT Books.BID,Books.Bname,Lending.L_Date  
FROM Books,Lending  
WHERE Books.BID=Lending.BID
```

```
Select * From view_1
```

Results		Messages	
BID	Bname	L_Date	
2	Access	2010-03-03	
1	Internet	2010-03-10	
5	Microsoft Access	2010-03-12	
3	Database	2010-03-03	
2	Access	2010-03-20	
5	Microsoft Access	2010-03-20	

♦ نام کتاب هایی که تعدادشان بین ۴ تا ۷ است را لیست کنید

♦ لیست اعضا را بر حسب حروف الفبا مرتب کنید

♦ تعداد برگرداندن کتاب ها را لیست کنید

♦ اسم نویسندگانی را نمایش دهید که ارزش مجموعه کتاب های آنها بیشتر از صد هزار تومان است

♦ نام یکتای کتاب هایی که حداقل یک بار قرض گرفته شده اند