

پایگاه داده ۲
جلسه ۶

فرجیان



IASBS
1992-2012

پایگاه داده‌ها

زبان پرس و جوی SQL



عملگرهاي ALL و ANY

- عملگر ALL برابر مقایسه "همه مقادیر" و عملگر ANY (SOME) برای "هر یک از مقادیر" تعریف شده‌اند.

- **مثال:** دانشجویانی که معدل آنها از همه دانشجویان دانشکده 10 بیشتر است.

```
SELECT *
```

```
FROM stud
```

```
WHERE avg > ALL ( SELECT avg FROM stud  
WHERE clg# = 10)
```



ALL

Name	Salary
Mary	20,000
John	15,000
Jane	25,000
Paul	30,000

Find the names of the employee(s) who earn the highest salary

```
SELECT Name
FROM Employee
WHERE Salary >=
  ALL (
    SELECT Salary
    FROM Employee)
```



ANY

Name	Salary
Mary	20,000
John	15,000
Jane	25,000
Paul	30,000

Find the names of employee(s) who earn more than someone else

```
SELECT Name
      FROM Employee
WHERE Salary >
      ANY (
          SELECT Salary
          FROM Employee)
```



مثال: دانشکده‌هایی که کد آنها عددی فرد و یک رقمی است.

IASBS
1992-2012

```
select *
```

```
from clg
```

```
where clg# = ANY(1 , 3 , 5 , 7 , 9)
```



IASBS
1992-2012

پرس و جو با قید ”همه“

- معادلسازی عملگر تقسیم در جبر رابطه‌ای با استفاده از توابع COUNT و یا CONTAINS امکان پذیر است.



مثال: شماره دانشجویانی که همه درس‌های دانشکده‌اشان را گرفته‌اند.

IASBS
1992-2012

```
SELECT stud.s#  
FROM stud,sec,crs  
WHERE stud.s#=sec.s#  
AND sec.c#=crs.c#  
GROUP BY stud.s#  
HAVING COUNT(DISTINCT sec.c#) = (SELECT COUNT(c#)  
FROM crs  
WHERE crs.clg#=stud.clg#);
```




مثال: شماره دانشجویانی که تمامی دروس ارایه شده توسط استاد تبریزی را گرفته‌اند.

IASBS
1992-2012

```
SELECT sec.s#  
FROM sec  
WHERE pname='تبریزی'  
GROUP BY sec.s#  
HAVING c# CONTAINS ( SELECT c#  
FROM sec  
WHERE pname='تبریزی')
```



join

- کاربرد: پیوند دادن جداول
- انواع:

JOIN ➤

(LEFT - RIGHT) OUTER JOIN ➤

FULL JOIN ➤

CROSS JOIN ➤



NATURAL JOIN

Student

Name	ID
John	123
Mary	124
Mark	125
Jane	126

Enrolment

Code	ID
DBS	123
PRG	124
DBS	124
PRG	126

SELECT * FROM

**Student NATURAL JOIN
Enrolment**

ID	Name	Code
123	John	DBS
124	Mary	PRG
124	Mary	DBS
126	Jane	PRG



Inner join

فرم کلي دستور:

```
SELECT <select list>  
FROM <first_table>  
<join_type> <second_table>  
[ON <join_condition>]
```



INNER JOIN

Buyer

Name	Budget
Smith	100,000
Jones	150,000
Green	80,000

Property

Address	Price
15 High St	85,000
12 Queen St	125,000
87 Oak Row	175,000

```
SELECT * FROM  
Buyer INNER JOIN  
Property ON  
Price <= Budget
```

Name	Budget	Address	Price
Smith	100,000	15 High St	85,000
Jones	150,000	15 High St	85,000
Jones	150,000	12 Queen St	125,000



مثال: اسامي اساتيد دانشكده كامپيوتر

IASBS
1992-2012

```
SELECT prof.pname  
FROM prof  
      JOIN clg  
      ON prof.clg#=clg.clg#  
WHERE clgname='كامپيوتر'
```

```
SELECT prof.pname  
FROM prof, clg  
WHERE prof.clg#=clg.clg#  
      AND clgname='كامپيوتر'
```



OUTER JOIN

ساختار کلي:

SELECT <SELECT list>

FROM <the table you want to be the "LEFT" table>

<LEFT|RIGHT> [OUTER] JOIN <table you want to be the
"RIGHT" table>

ON <join condition>

- استفاده از کلمه OUTER اختياري است.
- یک LEFT OUTER JOIN شامل همه اطلاعات جدول سمت چپ مي باشد.
- یک RIGHT OUTER JOIN شامل همه اطلاعات جدول سمت راست مي باشد.



مثال: نام تمام دروس، به همراه نام دانشکده ارائه دهنده آن.

IASBS
1992-2012

```
SELECT cname, clgname  
FROM crs  
LEFT OUTER JOIN clg  
ON crs.clg# = clg.clg#
```




مثال: نام تمام دانشکده‌ها، به‌مراه نام دروسی که ارائه می‌دهند.

I A S B S
1992 - 2012

```
SELECT cname, clgname
FROM crs RIGHT OUTER JOIN clg
ON crs.clg# = clg.clg#
```



FULL OUTER JOIN

IASBS
1992-2012

- وقتی می‌خواهیم همزمان LEFT JOIN و RIGHT JOIN را با هم داشته باشیم از FULL JOIN استفاده می‌نماییم.



Cross join

- تمام رکوردهای جدول سمت چپ را با تمام رکوردهای جدول سمت راست پیوند می‌دهد. (مشابه ضرب دکارتی)
- Cross join عملگر on ندارد.

- ساختار کلی:

```
SELECT <select list>  
FROM <first_table>  
<cross join> <second_table>
```



CROSS JOIN

IASBS
1992-2012

Student

Name	ID
John	123
Mary	124
Mark	125
Jane	126

Enrolment

Code	ID
DBS	123
PRG	124
DBS	124
PRG	126

```
SELECT * FROM  
Student CROSS JOIN  
Enrolment
```

Code	ID	Name	ID
DBS	123	John	123
DBS	123	Mary	124
DBS	123	Mark	125
DBS	123	Jane	126
PRG	124	John	123
PRG	124	Mary	124
PRG	124	Mark	125
PRG	124	Jane	126
DBS	124	John	123
DBS	124	Mary	124



CROSS and NATURAL JOIN

IASBS
1992-2012

```
SELECT * FROM  
  A CROSS JOIN B
```

- is the same as

```
SELECT * FROM A, B
```

```
SELECT * FROM  
  A NATURAL JOIN B
```

- is the same as

```
SELECT A.col1,... A.coln,  
[and all other columns  
apart from B.col1,...B.coln]  
FROM A, B  
WHERE A.col1 = B.col1  
      AND A.col2 = B.col2  
      ...AND A.coln = B.col.n  
(this assumes that col1...  
coln in A and B have  
common names)
```



انواع جداول در SQL

- **جدول اصلي (base table)**: با دستور `create table` ايجاد مي شود.
- **جدول مياني (intermediate table)**: توسط سيستم ايجاد مي شود.
- **جدول مجازي (views)**: وجود خارجي و فيزيكي ندارد اما مي توان به آن دسترسي داشت و آنرا تغيير داد.



جدول مجازي

- هدف جداول مجازي، ايجاد جدول خلاصه از اطلاعات موجود و محدود کردن دید کاربران است.

- شکل دستور:

```
CREATE VIEW view_name [(attributes)] AS  
SELECT...
```



مثال: جدول مجازي شامل نام، گروه‌ها و استاد درس.

IASBS
1992-2012

```
Create view v_tablae as  
Select cname, sec#, pname  
From crs,sec  
Where crs.c#=sec.c#
```

- **نکته:** مي‌توان ستون‌هاي جدول مجازي را با نام‌هاي جديد نام‌گذاري کرد.

```
Create view v_tablae (course , section , professor) as  
Select cname, sec#, pname  
From crs,sec  
Where crs.c#=sec.c#
```




دستورات کنترل مجوز دسترسي

- دستورات کنترل مجوز دسترسي در پایگاه داده به دو دسته تقسیم می‌شود:
 - دسترسي به داده ها
 - دسترسي به امکانات مدیریتی
- در اینجا دسترسي به داده‌ها بحث می‌شود.
 - دستور اعطاء اختیارات
 - دستور لغو اختیارات



دستور اعطاء اختيارات

IASBS
1992-2012

- شکل کلي دستور:

GRANT Privileges | ALL Privileges
ON Object
TO Users [WITH GRANT OPTION];

- در قسمت Privileges از يکي يا ترکيبي از کلمات DELETE، INSERT، UPDATE، SELECT استفاده مي‌گردد.
- در قسمت Object يکي از اشياء موجود در بانک انتخاب مي‌گردد.
- در قسمت Users نام يکي از کاربران پايگاه داده انتخاب مي‌گردد.
- اگر از WITH GRANT OPTION استفاده شود، يعني کاربر مي‌تواند اين امتيازها را به کاربران ديگر نيز واگذار نمايد.



مثال:

IASBS
1992-2012

```
GRANT INSERT, SELECT, UPDATE ON crs, sec  
TO haghjoo, rahimi;
```

```
GRANT UPDATE(degree, esp) ON prof  
TO naime;
```

```
GRANT SELECT ON university  
TO Rahimi WITH GRANT OPTION;
```



دستور لغو اختیارات

- برای لغو امتیاز از دستور **REVOKE** استفاده می‌گردد.
- شکل کلی دستور:

```
REVOKE Privileges | ALL Privileges  
ON Object  
FROM Users
```

- مثال:

```
REVOKE INSERT ON crs, sec  
FROM haghjoo;
```



IASBS
1992 - 2012

پایان جلسه ششم