Intermediate SQL

قمرناز تدین

عبارت های الحاق شده

- **عملگرهای join** دو رابطه را دریافت می کنند و به عنوان نتیجه، یک رابطه دیگر را برمیگردانند. عملگر join درحقیقت ضرب کارتزین است که در آن باید رکوردهای دو رابطه تحت شرایط خاصی با هم منطبق شوند. همچنین شامل خصیصه های والد می باشد.
- عملیات الحاق معمولا به عنوان زیرپرس و جو در بخش عبارت from استفاده می شوند.

عمبگرهای Join - مثال

- Relation *course*

course_id	*title*	*dept_name*	*credits*
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

- Relation *prereq*

course_id	*prereq_id*
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

- مشاهده می شود که:

اطلاعات prereq برای CS-315 و اطلاعات course برای CS-347 وجود ندارند.

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

The *student* relation.

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	*null*

The *takes* relation.

select *student.ID* **as** *ID*, *name*, *dept _name*, *tot_cred*,*course_id*, *sec_id*, *semester*, *year*, *grade*
from *student* **join** *takes* **on** *student.ID= takes.ID*;

ID	name	dept_name	tot_cred	course_id	sec_id	semester	year	grade
00128	Zhang	Comp. Sci.	102	CS-101	1	Fall	2009	A
00128	Zhang	Comp. Sci.	102	CS-347	1	Fall	2009	A-
12345	Shankar	Comp. Sci.	32	CS-101	1	Fall	2009	C
12345	Shankar	Comp. Sci.	32	CS-190	2	Spring	2009	A
12345	Shankar	Comp. Sci.	32	CS-315	1	Spring	2010	A
12345	Shankar	Comp. Sci.	32	CS-347	1	Fall	2009	A
19991	Brandt	History	80	HIS-351	1	Spring	2010	B
23121	Chavez	Finance	110	FIN-201	1	Spring	2010	C+
44553	Peltier	Physics	56	PHY-101	1	Fall	2009	B-
45678	Levy	Physics	46	CS-101	1	Fall	2009	F
45678	Levy	Physics	46	CS-101	1	Spring	2010	B+
45678	Levy	Physics	46	CS-319	1	Spring	2010	B
54321	Williams	Comp. Sci.	54	CS-101	1	Fall	2009	A-
54321	Williams	Comp. Sci.	54	CS-190	2	Spring	2009	B+
55739	Sanchez	Music	38	MU-199	1	Spring	2010	A-
76543	Brown	Comp. Sci.	58	CS-101	1	Fall	2009	A
76543	Brown	Comp. Sci.	58	CS-319	2	Spring	2010	A
76653	Aoi	Elec. Eng.	60	EE-181	1	Spring	2009	C
98765	Bourikas	Elec. Eng.	98	CS-101	1	Fall	2009	C-
98765	Bourikas	Elec. Eng.	98	CS-315	1	Spring	2010	B
98988	Tanaka	Biology	120	BIO-101	1	Summer	2009	A
98988	Tanaka	Biology	120	BIO-301	1	Summer	2010	*null*

الحاق خارجی Outer Join

- توسعه ای از عملگر الحاق که از ازدست دادن اطلاعات جلوگیری می کند.
- الحاق را محاسبه میکند و سپس رکوردهایی از یک رابطه را که با رکوردهای رابطه دیگر منطبق نشده اند به نتیجه الحاق اضافه می کند.
- از مقادیر null استفاده می کند.

Left Outer Join

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

The *student* relation.

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	*null*

The *takes* relation.

ID	name	dept_name	tot_cred	course_id	sec_id	semester	year	grade
00128	Zhang	Comp. Sci.	102	CS-101	1	Fall	2009	A
00128	Zhang	Comp. Sci.	102	CS-347	1	Fall	2009	A-
12345	Shankar	Comp. Sci.	32	CS-101	1	Fall	2009	C
12345	Shankar	Comp. Sci.	32	CS-190	2	Spring	2009	A
12345	Shankar	Comp. Sci.	32	CS-315	1	Spring	2010	A
12345	Shankar	Comp. Sci.	32	CS-347	1	Fall	2009	A
19991	Brandt	History	80	HIS-351	1	Spring	2010	B
23121	Chavez	Finance	110	FIN-201	1	Spring	2010	C+
44553	Peltier	Physics	56	PHY-101	1	Fall	2009	B-
45678	Levy	Physics	46	CS-101	1	Fall	2009	F
45678	Levy	Physics	46	CS-101	1	Spring	2010	B+
45678	Levy	Physics	46	CS-319	1	Spring	2010	B
54321	Williams	Comp. Sci.	54	CS-101	1	Fall	2009	A-
54321	Williams	Comp. Sci.	54	CS-190	2	Spring	2009	B+
55739	Sanchez	Music	38	MU-199	1	Spring	2010	A-
70557	Snow	Physics	0	*null*	*null*	*null*	*null*	*null*
76543	Brown	Comp. Sci.	58	CS-101	1	Fall	2009	A
76543	Brown	Comp. Sci.	58	CS-319	2	Spring	2010	A
76653	Aoi	Elec. Eng.	60	EE-181	1	Spring	2009	C
98765	Bourikas	Elec. Eng.	98	CS-101	1	Fall	2009	C-
98765	Bourikas	Elec. Eng.	98	CS-315	1	Spring	2010	B
98988	Tanaka	Biology	120	BIO-101	1	Summer	2009	A
98988	Tanaka	Biology	120	BIO-301	1	Summer	2010	*null*

select * **from** *student* **natural left outer join** *takes*;

Left Outer Join

- Relation *course*

course_id	*title*	*dept_name*	*credits*
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

- Relation *prereq*

course_id	*prereq_id*
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

- *course* **natural left outer join** *prereq*

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*

Right Outer Join

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

The *student* relation.

ID	course_id	sec_id	semester	year	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	*null*

The *takes* relation.

select *
from *takes* **natural right outer join** *student*;

ID	course_id	sec_id	semester	year	grade	name	dept_name	tot_cred
00128	CS-101	1	Fall	2009	A	Zhang	Comp. Sci.	102
00128	CS-347	1	Fall	2009	A-	Zhang	Comp. Sci.	102
12345	CS-101	1	Fall	2009	C	Shankar	Comp. Sci.	32
12345	CS-190	2	Spring	2009	A	Shankar	Comp. Sci.	32
12345	CS-315	1	Spring	2010	A	Shankar	Comp. Sci.	32
12345	CS-347	1	Fall	2009	A	Shankar	Comp. Sci.	32
19991	HIS-351	1	Spring	2010	B	Brandt	History	80
23121	FIN-201	1	Spring	2010	C+	Chavez	Finance	110
44553	PHY-101	1	Fall	2009	B-	Peltier	Physics	56
45678	CS-101	1	Fall	2009	F	Levy	Physics	46
45678	CS-101	1	Spring	2010	B+	Levy	Physics	46
45678	CS-319	1	Spring	2010	B	Levy	Physics	46
54321	CS-101	1	Fall	2009	A-	Williams	Comp. Sci.	54
54321	CS-190	2	Spring	2009	B+	Williams	Comp. Sci.	54
55739	MU-199	1	Spring	2010	A-	Sanchez	Music	38
70557	*null*	*null*	*null*	*null*	*null*	Snow	Physics	0
76543	CS-101	1	Fall	2009	A	Brown	Comp. Sci.	58
76543	CS-319	2	Spring	2010	A	Brown	Comp. Sci.	58
76653	EE-181	1	Spring	2009	C	Aoi	Elec. Eng.	60
98765	CS-101	1	Fall	2009	C-	Bourikas	Elec. Eng.	98
98765	CS-315	1	Spring	2010	B	Bourikas	Elec. Eng.	98
98988	BIO-101	1	Summer	2009	A	Tanaka	Biology	120
98988	BIO-301	1	Summer	2010	*null*	Tanaka	Biology	120

Right Outer Join

- Relation *course*

course_id	*title*	*dept_name*	*credits*
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

- Relation *prereq*

course_id	*prereq_id*
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

- *course* **natural right outer join** *prereq*

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-347	*null*	*null*	*null*	CS-101

Joined Relations

- **عملگرهای الحاق** دو رتبطه را میگیرند و یک رابطه را برمیگردانند.
- **شرط الحاق-** تعریف میکند که کدام رکوردهای دو رابطه منطبق شوند و کدام خصیصه ها در نتیجه الحاق وجود داشته باشند.
- **نوع الحاق-** تعریف میکند که وضعیت رکوردهای هر رابطه که با هیچ رکوردی از رابطه دیگر منطبق نیستند چگونه باشد.

Join types
inner join
left outer join
right outer join
full outer join

Join Conditions
natural
on <predicate>
using $(A_1, A_1, \ldots, A_n)$

Full Outer Join

- Relation *course*

course_id	*title*	*dept_name*	*credits*
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

- Relation *prereq*

course_id	*prereq_id*
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

- *course* **natural full outer join** *prereq*

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*
CS-347	*null*	*null*	*null*	CS-101

مثال -Joined Relations

- *course* **inner join** *prereq* **on** *course.course_id = prereq.course_id*

course_id	*title*	*dept_name*	*credits*	*prereq_id*	*course_id*
BIO-301	Genetics	Biology	4	BIO-101	BIO-301
CS-190	Game Design	Comp. Sci.	4	CS-101	CS-190

- تفاوت بین رابطه بالا و الحاق طبیعی چیست؟

- *course* **left outer join** *prereq* **on** *course.course_id = prereq.course_id*

course_id	*title*	*dept_name*	*credits*	*prereq_id*	*course_id*
BIO-301	Genetics	Biology	4	BIO-101	BIO-301
CS-190	Game Design	Comp. Sci.	4	CS-101	CS-190
CS-315	Robotics	Comp. Sci.	3	*null*	*null*

Joined Relations – Examples

- *course* **natural right outer join** *prereq*

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-347	*null*	*null*	*null*	CS-101

- *course* **full outer join** *prereq* **using** (*course_id*)

course_id	*title*	*dept_name*	*credits*	*prereq_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*
CS-347	*null*	*null*	*null*	CS-101

دیدگاهها Views

- گاهی مطلوب نیست که همه کاربران بتوانند همه مدل منطقی را ببینند. (منظور، رابطه واقعی است که در پایگاه داده ذخیره شده)
- فرض کنید میخواهیم شماره، نام و گروه استادها را ببینیم. بنابراین پرس و جو به شکل زیر است:

```
select ID, name, dept_name
from instructor
```

- **دیدگاه** مکانیزمی برای مخفی کردن بخشی از اطلاعات از دید کاربران معینی اس.
- هر رابطه ای که یک مدل مفهومی نباشد ولی به صورت «رابطه مجازی» برای کاربر قابل مشاهده باشد یک دیدگاه گفته می شود.

تعریف View

- دیدگاه به صورت زیر تعریف می شود:

create view *v* **as** < query expression >

که <query expression> هر عبارت SQL می تواند باشد. V نام دیدگاه است.

- زمانی که دیدگاه تعریف می شود، نام آن برای ارجاع به یک رابطه مجازی که دیدگاه تولید می کند استفاده می شود.
- تعریف دیدگاه مشابه ایجاد رابطه جدید نیست.
  - تعریف دیدگاه، باعث می شود که فقط یک عبارت ذخیره شود و این رابطه به پرس و جوهایی که برروی دیدگاه اجرا میشوند، اضافه می شود.

مثالهایی از Views

- دیدگاهی از استادان بدون حقوق آنها:

```
create view faculty as
   select ID, name, dept_name
   from instructor
```

- همه استادهای گروه Biology

```
select name
from faculty
where dept_name = 'Biology'
```

- ایجاد دیدگاهی براساس جمع حقوق هر گروه

```
create view departments_total_salary(dept_name, total_salary) as
     select dept_name, sum (salary)
     from instructor
    group by dept_name;
```

تعریف دیدگاه برروی دیدگاههای دیگر

- **create view** *physics_fall_2009* **as**
  select *course.course_id*, *sec_id*, *building*, *room_number*
  from *course*, *section*
  where *course.course_id* = *section.course_id*
  and *course.dept_name* = 'Physics'
  and *section.semester* = 'Fall'
  and *section.year* = '2009';
- **create view** *physics_fall_2009_watson* **as**
  select *course_id*, *room_number*
  from *physics_fall_2009*
  where *building*= 'Watson';

تراکنش ها Transactions

- واحد انجام کار
- تراکنش اتمیک Atomic transaction
  - یا به طور کامل اجرا می شود یا به طور کامل کنسل می شود مثل این که اصلا اجرا نشده است.
- جداسازی (Isolation) از تراکنش های موازی.
- تراکنش ها به صورت ضمنی شروع می شوند.
  - با یکی از این عبارتها خاتمه می یابند:

commit work or **rollback work**

ایجاد ایندکس

- **create table** *student*
  (*ID* **varchar** (5),
  name **varchar** (20) **not null**,
  dept_name **varchar** (20),
  tot_cred **numeric** (3,0) **default** 0,
  primary key (*ID*))
- **create index** *studentID_index* **on** *student*(*ID*)
  - ایندکسها ساختارهای داده ای هستند که برای سرعت بخشیدن به دستیابی به رموردهایی با مقادیر معین برای خصیصه های ایندکس می شود.
    - **select ***
      from *student*
      where *ID* = ‘12345’

این پرس و جو می تواند با استفاده از ایندکس برای یافتن رکود موردنظر اجرا شود، بدون نیاز به این که همه رکوردهای student جستجو شوند.

Data Types in SQL

- **date**: تاریخ، شامل سال (چهار رقمی)، ماه و روز
  - Example: **date** ‘2005-7-27’
- **Time** : زمان شامل ساعت، دقیقه و ثانیه
  - Example: **time** ‘09:00:30’
  - **Timestamp**: تاریخ به اضافه ساعت
  - Example: **timestamp** ‘2005-7-27 09:00:30.75’

تعریف نوع داده توسط کاربر

create type *Dollars* **as numeric (12,2) final**

- **create table** *department*
  (*dept_name* **varchar** (20),
  building **varchar** (15),
  budget Dollars);

Domains

create domain *person_name* **char**(20) **not null**

- Type و domain مشابهند ولی در domain می توان محدودیتهایی مثل **not null** هم تعریف کرد.
- **create domain** *degree_level* **varchar**(10)
  constraint *degree_level_test*
  check (**value in** (’Bachelors’, ’Masters’, ’Doctorate’));

مجوزدهی

شکلهای مجوزدهی در بخش های مختلف پایگاه داده:

- **Read:** خواندن داده ها مجاز است ولی تغییر در آنها مجاز نیست.
- **Insert:** درج داده جدید مجاز است ولی تغییر در داده موجود مجاز نیست.
- **Update:** تغییر در داده ها مجاز است ولی حذف آنها مجاز نیست.
- **Delete:** حذف داده ها مجاز است.

شکل های مجوزدهی برای تغییر در شمای پایگاه داده ها:

- **Index**- ایجاد و حذف ایندکس مجاز است.
- **Resources**- ایجاد رابطه های جدید مجاز است.
- **Alteration** اضافه کردن یا حذف خصیصه های رابطه مجاز است.
- **Drop**- حذف رابطه مجاز است.

تعیین مجوزها در SQL

- عبارت grant برای تعریف مجوزدهی ها استفاده می شود

grant <privilege list>

on <relation name or view name> **to** <user list>

- <user list>میتواند یکی از موارد زیر باشد:
  - user-id
  - **public**، که به معنی همه کاربران است.
- اگر مجوزی بر روی یک دیدگاه داده شود، این مجوز بر رابطه ها اعمال نمی شود.
- کاربری که مجوزی را صادر میکند خودش باید دارای آن مجوز باشد (یا متصدی پایگاه داده باشد)

مجوزها در SQL

- **select**: دستیابی خواندن بر یک رابطه یا امکان پرس و جو با استفاده از دیدگاه.
  - مثال: به کاربران U_1, U_2و U_3 مجوز select در رابطه *instructor* داده شود.

    grant select on *instructor* **to** U_1, U_2, U_3
- **Insert** : قابلیت درج رکوردها.
- **Update**: قابلیت بروزرسانی رکوردها.
- **Delete**: قابلیت حذف رکوردها.
- **all privileges**: همه مجوزهای ممکن.

لغو مجوزها در SQL

- عبارت revoke برای لغو مجوز استفاده می شود.

revoke <privilege list>
on <relation name or view name> **from** <user list>

- مثال:

revoke select on *instructor* **from** U_1, U_2, U_3

- اگر <privilege-list> برابر با all باشد، همه مجوزها لغو می شوند.
- اگر <revokee-list> برابر با public باشد همه کاربران مجوزشان لغو میشود.