



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

برنامه سازی پیشرفته و کارگاه

دیتابیس مقدماتی ۲

استاد درس

دکتر مهدی قطعی

استاد دوم

بهنام یوسفی مهر

نگارش

سیدآرمان حسینی

بهار ۱۴۰۳

joinها

foreign keyها رو یاد تونه؟ بهتون اجازه می‌داده بین tableهای مختلف، ارتباط برقرار کنید. یکی از دستوراتی که بهتون اجازه می‌ده از این foreign keyها استفاده کنید، دستورات خانواده join و توی این بخش، می‌خوایم با اون‌ها به خورده‌ور بریم.

اولین join

کوئری‌هایی که تا حالا زدین، فقط با recordهای به جدول درگیر بودن. مثلاً کوئری زیر، فقط رکوردهای جدول tasks رو خروجی می‌داد:

```
select id, employee_id, name from tasks;
```

یا دستور زیر، فقط stepهای تسک شماره ۳ رو نشون می‌داد:

```
select name from steps
where task_id = 3;
```

حالا فرض کنید که بخوایم توی به کوئری، هم اطلاعات taskها رو نشون بدیم، هم اطلاعات stepها رو. چه کار می‌کنیم؟

این جاست که دستور join به کارمون می‌اد. دستور join، ستون‌های دو یا چند جدول رو با هم ترکیب می‌کنه. به عنوان مثال، دستور زیر رو توی consoleتون اجرا کنید:

```
select * from tasks
join steps on tasks.id = steps.task_id;
```

این دستور، به mysql همچنین چیزی می‌گه:

1. اول، همه رکوردهای جدول tasks رو بخون.
2. بعدش، همه رکوردهای جدول steps رو بخون.
3. حالا، هر step رو، کنار taskای نشون بده که idی اون task، برابر task_idی step نمونه.

یه نگاه به خروجی این دستور بندازید. خروجی‌ش، به table گنده‌ست که همه ستون‌های جداول tasks و steps رو داره. این table، به همچنین شکلی داره:

tasks.id	task_list_id	employee_id	tasks.name	due_date	steps.id	task_id	steps.name	is_completed
1	1	<null>	1 Meeting with costumers	2025-12-13	1	1	1 Set the meeting	1
2	1	<null>	1 Meeting with costumers	2025-12-13	2	2	1 Attend the meeting	0
3	3	1	2 Develop "task completion" feature	2025-12-14	3	3	3 Study related use cases	1
4	3	1	2 Develop "task completion" feature	2025-12-14	4	4	3 Implementation	1
5	4	1	4 Develop "tag" feature	2025-12-15	5	5	4 Study related use cases	1
6	4	1	4 Develop "tag" feature	2025-12-15	6	6	4 Implementation	0
7	6	2	3 Test "task completion" feature	2025-12-20	7	7	6 Review possible scenarios	0
8	6	2	3 Test "task completion" feature	2025-12-20	8	8	6 Testing	0
9	7	2	3 Test "tag" feature	2025-12-20	9	9	7 Review possible scenarios	1
10	7	2	3 Test "tag" feature	2025-12-20	10	7	7 Testing	0

خوندن اطلاعات جدول بالا سخته، بیاین توی selectمون، از سر یه سری از ستون‌هاش خلاص بشیم تا بتونیم بهتر بررسیش کنیم:

```
select tasks.id, tasks.name, steps.id, task_id, steps.name from tasks
join steps on tasks.id = steps.task_id;
```

اگر دوباره این دستور رو اجرا کنید، خروجی زیر رو می‌بینید:

tasks.id	tasks.name	steps.id	task_id	steps.name
1	1 Meeting with costumers	1	1	1 Set the meeting
2	1 Meeting with costumers	2	2	1 Attend the meeting
3	3 Develop "task completion" feature	3	3	3 Study related use cases
4	3 Develop "task completion" feature	4	4	3 Implementation
5	4 Develop "tag" feature	5	5	4 Study related use cases
6	4 Develop "tag" feature	6	6	4 Implementation
7	6 Test "task completion" feature	7	7	6 Review possible scenarios
8	6 Test "task completion" feature	8	8	6 Testing
9	7 Test "tag" feature	9	9	7 Review possible scenarios
10	7 Test "tag" feature	10	7	7 Testing

همون‌طور که می‌بینید، توی جدول بالا هر step به همراه task مربوط بهش دیده می‌شه. بعضی از taskها، دو بار توی خروجی دیده می‌شن، چون که دوتا step داشتن! ولی بعضی taskها هم اصلاً توی این جدول دیده نمی‌شن، چون به کل step نداشتن.

اگر به جدول بالا دقت کنید، یه چیز جالب توی اون می‌بینید. این که ستون‌های id، به عنوان tasks.id و steps.id نمایش داده شدن. فرض کنید mysql این کار رو نمی‌کرد:

id	name	id	task_id	name
1	1 Meeting with costumers	1	1	1 Set the meeting
2	1 Meeting with costumers	2	2	1 Attend the meeting
3	3 Develop "task completion" feature	3	3	3 Study related use cases
4	3 Develop "task completion" feature	4	4	3 Implementation
5	4 Develop "tag" feature	5	5	4 Study related use cases
6	4 Develop "tag" feature	6	6	4 Implementation
7	6 Test "task completion" feature	7	7	6 Review possible scenarios
8	6 Test "task completion" feature	8	8	6 Testing
9	7 Test "tag" feature	9	9	7 Review possible scenarios
10	7 Test "tag" feature	10	7	7 Testing

توی جدول بالا، کدوم ستون id مربوط به task و کدوم مربوط به step؟ ستون name چطور؟

در واقع، چون این ستون ها توی هر دو جدول مشترکن، اسم جدول اون ها هم بالای ستونشون اومده. اگر دقت کنید، حتی توی کوئری مون هم هر جایی ما از این دو ستون استفاده کردیم، مجبور شدیم قبل اون ها اسم جدولشون رو بنویسیم:

```
select tasks.id, tasks.name, steps.id, task_id, steps.name from tasks
join steps on tasks.id = steps.task_id;
```

ما می تونستیم این کوئری رو کوتاه تر هم بنویسیم:

```
select t.id, t.name, s.id, task_id, s.name from tasks t
join steps s on t.id = s.task_id;
```

توی کوئری جدیدمون، به جای tasks از t و به جای steps از s استفاده کردیم. برای این که mysql بدون منظرمون از t و s چیه، جلوی اسم این جداول هر کدوم رو مشخص کردیم. خروجی این کوئری هم مثل کوئری قبلیه، با این تفاوت که mysql هم از t و s استفاده کرده:

	t.id	t.name	s.id	task_id	s.name
1	1	Meeting with costumers	1	1	Set the meeting
2	1	Meeting with costumers	2	2	Attend the meeting
3	3	Develop "task completion" feature	3	3	Study related use cases
4	3	Develop "task completion" feature	4	4	Implementation
5	4	Develop "tag" feature	5	5	Study related use cases
6	4	Develop "tag" feature	6	6	Implementation
7	6	Test "task completion" feature	7	7	Review possible scenarios
8	6	Test "task completion" feature	8	8	Testing
9	7	Test "tag" feature	9	9	Review possible scenarios
10	7	Test "tag" feature	10	7	Testing

left join

اگر بخوایم task هایی که step ای ندارن هم توی خروجی بالا نمایش داده بشن، از left join استفاده می کنیم:

```
select t.id, t.name, s.id, task_id, s.name from tasks t
left join steps s on t.id = s.task_id;
```

خروجی دستور بالا، به این شکله:

	t.id	t.name	s.id	task_id	s.name
1	1	Meeting with costumers	1	1	Set the meeting
2	1	Meeting with costumers	2	1	Attend the meeting
3	2	Fix bathroom	<null>	<null>	<null>
4	3	Develop "task completion" feature	3	3	Study related use cases
5	3	Develop "task completion" feature	4	3	Implementation
6	4	Develop "tag" feature	5	4	Study related use cases
7	4	Develop "tag" feature	6	4	Implementation
8	5	Fix "task lists" bug	<null>	<null>	<null>
9	6	Test "task completion" feature	7	6	Review possible scenarios
10	6	Test "task completion" feature	8	6	Testing
11	7	Test "tag" feature	9	7	Review possible scenarios
12	7	Test "tag" feature	10	7	Testing
13	8	Test "task lists" bug	<null>	<null>	<null>

همون طور که می بینید، task های بدون step هم توی کوئری بالا نمایش داده شدن. توی سطرهای این task ها ستون های s.id، task_id و s.name مقدار null رو گرفتن.

کوئری زدن روی نتایج

ما می تونیم مثل جدول های قبلی مون، روی این جدول هم where و group by و کوئری های مختلف بزنینم. مثلاً توی کوئری زیر، task هایی که هیچ step ای ندارن رو نشون می دیم:

```
select t.id, t.name, s.id from tasks t
left join steps s on t.id = s.task_id
where s.id is null;
```

خروجی این دستور، به همین شکلیه:

	t.id	name	s.id
1	2	Fix bathroom	<null>
2	5	Fix "task lists" bug	<null>
3	8	Test "task lists" bug	<null>

یا می تونیم ببینیم که هر task چند step داره:

```
select t.id, t.name, count(s.id) as number_of_steps from tasks t
left join steps s on t.id = s.task_id
group by t.id;
```

خروجی دستور بالا به شکل زیره:

	id	name	number_of_steps
1	1	Meeting with costumers	2
2	2	Fix bathroom	0
3	3	Develop "task completion" feature	2
4	4	Develop "tag" feature	2
5	5	Fix "task lists" bug	0
6	6	Test "task completion" feature	2
7	7	Test "tag" feature	2
8	8	Test "task lists" bug	0

حتی می‌تونیم ببینیم که از هر task، چند step تموم شده:

```
select
  t.id,
  t.name,
  count(s.id) as number_of_steps,
  sum(s.is_completed = true) as completed_steps
from tasks t
left join steps s on t.id = s.task_id
group by t.id;
```

خروجی کد بالا هم به شکل زیره:

	id	name	number_of_steps	completed_steps
1	1	Meeting with costumers	2	1
2	2	Fix bathroom	0	<null>
3	3	Develop "task completion" feature	2	2
4	4	Develop "tag" feature	2	1
5	5	Fix "task lists" bug	0	<null>
6	6	Test "task completion" feature	2	0
7	7	Test "tag" feature	2	1
8	8	Test "task lists" bug	0	<null>

همون طور که می‌بینید، ستون completed_steps برای task‌های بدون step به جای 0، null گرفته. برای حل این مشکل از ifnull استفاده می‌کنیم. اگر کد زیر براتون واضح نیست، یه سرچ کوچیک راجع به این تابع بکنید:

```
select
  t.id,
  t.name,
  count(s.id) as number_of_steps,
  ifnull(sum(s.is_completed = true), 0) as completed_steps
from tasks t
left join steps s on t.id = s.task_id
group by t.id;
```

join چند جدول

ما حتی می‌تونیم بیشتر از دو جدول رو با هم join کنیم. مثل بیاید جدول employee هم با task و step جوین کنیم:

```
select
  e.id as employee_id,
  e.name as employee_name,
  t.id as task_id,
  t.name as task_name,
  s.id as step_id,
  s.name as step_name
from employees e
left join tasks t on t.employee_id = e.id
left join steps s on s.task_id = t.id;
```

خروجی دستور بالا، به این شکله:

	employee_id	employee_name	task_id	task_name	step_id	step_name
1	1	Amin	1	Meeting with costumers	1	Set the meeting
2	1	Amin	1	Meeting with costumers	2	Attend the meeting
3	1	Amin	2	Fix bathroom	<null>	<null>
4	2	Ali	3	Develop "task completion" feature	3	Study related use cases
5	2	Ali	3	Develop "task completion" feature	4	Implementation
6	2	Ali	5	Fix "task lists" bug	<null>	<null>
7	3	Mamad	6	Test "task completion" feature	7	Review possible scenarios
8	3	Mamad	6	Test "task completion" feature	8	Testing
9	3	Mamad	7	Test "tag" feature	9	Review possible scenarios
10	3	Mamad	7	Test "tag" feature	10	Testing
11	3	Mamad	8	Test "task lists" bug	<null>	<null>
12	4	Torob	4	Develop "tag" feature	5	Study related use cases
13	4	Torob	4	Develop "tag" feature	6	Implementation

با استفاده از جدول بالا هم می‌تونیم کوئری‌های خوبی راجع به کارکنان شرکت بزنیم. مثلاً این که هر کارمند، چند قدم از کارهایش رو انجام داده:

```
select
  e.id as employee_id,
  e.name as employee_name,
  count(s.id) as total_number_of_steps,
  ifnull(sum(s.is_completed = true), 0) as completed_steps
from employees e
left join tasks t on t.employee_id = e.id
left join steps s on s.task_id = t.id
group by e.id;
```

خروجی این کوئری هم به شکل زیره:

	employee_id	employee_name	total_number_of_steps	completed_steps
1	1	Amin	2	1
2	2	Ali	2	2
3	3	Mamad	4	1
4	4	Torob	2	1