Cross Join

CROSS JOIN produces a result set which is the number of rows in the first table multiplied by the number of rows in the second table if no WHERE clause is used along with CROSS JOIN. This kind of result is called as Cartesian product.

Syntax

```
SELECT *
FROM table 1
CROSS JOIN table 2;
```

Simple Example:

Table 1 Teacher

Teacher_id	Teacher_name
1	Henry Li
2	Betty Wang
3	Leo Zhang

Table 2 Student

Teacher_id	Student_id	Student_name
1	5	Peter
2	6	Mary
3	7	Jerry

By using CROSS JOIN, we let each row from table 1 joins with all the rows of table 2. Because table 1 has three rows and table 2 has three rows, the result should return 3 * 3 = 9 rows below.

Results:

a.Teacher_id	a.Teacher_name	b.Teacher_id	b.Student_id	b.Student_name
1	Henry Li	1	5	Peter
1	Henry Li	2	6	Mary
1	Henry Li	3	7	Jerry
2	Betty Wang	1	5	Peter
2	Betty Wang	2	6	Mary
2	Betty Wang	3	7	Jerry
3	Leo Zhang	1	5	Peter
3	Leo Zhang	2	6	Mary
3	Leo Zhang	3	7	Jerry

Note: If WHERE clause is used with CROSS JOIN, it functions like an INNER JOIN.

For example, the following two syntax will provide the same results.

```
SELECT *
FROM teacher a
CROSS JOIN student b
Where a.teacher_id = b.teacher_id
```

```
SELECT *
FROM teacher a
INNER JOIN student b
ON a.teacher_id = b.teacher_id
```

However, in such case, INNER JOIN is more efficient than CROSS JOIN to achieve the same outcome as below.

a.Teacher_id	a.Teacher_name	b.Teacher_id	b.Student_id	b.Student_name
1	Henry Li	1	5	Peter
2	Betty Wang	2	6	Mary
3	Leo Zhang	3	7	Jerry

Self Join

Self JOIN is a regular join. It is used to join a table to itself as if the table were two tables. The syntax is the same as with any other regular joins such as INNER, or LEFT JOIN.

It is commonly used when we are working with a single table, which contains lots of information.

For example, a single table teacher-student:

Teacher_id	Teacher_name (tname)	Student_name (sname)
1	Henry Li	Peter
1	Henry Li	Kate
1	Henry Li	Brown
2	Betty Wang	Mary
3	Leo Zhang	Jerry

If we want to know the name of the students who share the same teacher, we can use self join. To make it easy to follow, we can do it in three steps:

Step 1: Self join on teacher_id

```
select *
from teacher_student a
join teacher_student b
on a.tname = b.tname
```

Step 2: remove same student joined to same student

```
select *
from teacher_student a
join teacher_student b
on a.tname = b.tname
and a.sname <> b.sname;
```

Step 3: remove duplicate student names

```
select distinct a.sname
from teacher_student a
join teacher_student b
on a.tname = b.tname
and a.sname <> b.sname;
```

Results

sname
Brown
Kate
Peter

Additional Exercise:

We have a table that tracks when a user opened an app. The sample data looks like this:

User_id	Date
1	2018-12-01
1	2018-12-03
1	2018-12-08
1	2018-12-20
2	2018-12-10
2	2018-12-20

Q1: We want to know when the user opened an app on a day, if the user also opened the app within 7 days before that date. This is defined as Weekly Active User (WAU).

For example, User 1 opened the app on 2018-12-03 and he/she also opened it on 2018-12-01 so he/she was WAU on 2018-12-03. However, he/she was not WAU on 2018-12-01 because he/she didn't open the app within 7 days before that date.

User 2 opened the app on 2018-12-10 and 2018-12-20 but he/she was not WAU on both dates because he/she didn't open the app within 7 days before both dates.

Can you use self join to return WAU of all users on dates when they opened the app? Please fill out the following table with SQL.

Tips: you should join on both user_id and date. **Synax: date - integer '7'** give the date 7 days ago.

User_id	Date	WAU
1	2018-12-01	No
1	2018-12-03	Yes
1	2018-12-08	Yes
1	2018-12-20	No
2	2018-12-10	No

2	2018-12-20	No

```
select  distinct a.id, a.date,
                case when b.id is null then 'No'
                else 'Yes'
                end as wau
from activity a
left join activity b
on a.User_id = b.User_id
and b.date > a.date - 7  -- join only with the dates within 7 days from the current
date
and b.date < a.date   -- don’t join with the current date
order by a.id, a.date;
```