

```

-- Here is another example that illustrates outer joins:

-- Let's start with two tables: Employee(id, name) and Sales(employeeID, productID)
-- The tables have the following content
-- (1,'Joe')          (1, 344)
-- (2,'Jack')         (1, 355)
-- (3,'Gill')         (2, 544)
-- If we run a simple join, Gill will not appear in the result
-- because she did not make any sells. This is the same reason why
-- our example with T intersect (R union S) would return the empty set
-- when any of the relations was empty: if a tuple does not have matches
-- in all of R, S, and T, it is not part of the result.
-- If we run a left outer-join, Gill will be returned with a null sale.
-- We can similarly do right outer joins and full outer joins (but not in sqlite)
-- Here is the example to run in sqlite:
create table Employee(id int, name varchar(10));
create table Sales(employeeID int, productID int);
insert into Employee values(1,'Joe');
insert into Employee values(2,'Jack');
insert into Employee values(3,'Gill');

insert into Sales values(1,344);
insert into Sales values(1,355);
insert into Sales values(2,544);

-- The following will miss Gill
select * from Employee E, Sales S where E.id = S.employeeID;

-- This is the same as above but with a different syntax:
select * from Employee E inner join Sales S on E.id = S.employeeID;

-- The following will include Gill
select * from Employee E left outer join Sales S on E.id = S.employeeID;

```