Can anyone give me a good example of when CROSS APPLY makes a difference in those cases where INNER JOIN will work as well?

See the article in my blog for detailed performance comparison:

* [**INNER JOIN vs. CROSS APPLY**](http://explainextended.com/2009/07/16/inner-join-vs-cross-apply/)

CROSS APPLY works better on things that have no simple JOIN condition.

This one selects 3 last records from t2 for each record from t1:

SELECT t1.\*, t2o.\*

FROM t1

CROSS APPLY

(

SELECT TOP 3 \*

FROM t2

WHERE t2.t1\_id = t1.id

ORDER BY

t2.rank DESC

) t2o

It cannot be easily formulated with an INNER JOIN condition.

You could probably do something like that using CTE's and window function:

WITH t2o AS

(

SELECT t2.\*, ROW\_NUMBER() OVER (PARTITION BY t1\_id ORDER BY rank) AS rn

FROM t2

)

SELECT t1.\*, t2o.\*

FROM t1

INNER JOIN

t2o

ON t2o.t1\_id = t1.id

AND t2o.rn <= 3

, but this is less readable and probably less efficient.

**Update:**

Just checked.

master is a table of about 20,000,000 records with a PRIMARY KEY on id.

This query:

WITH q AS

(

SELECT \*, ROW\_NUMBER() OVER (ORDER BY id) AS rn

FROM master

),

t AS

(

SELECT 1 AS id

UNION ALL

SELECT 2

)

SELECT \*

FROM t

JOIN q

ON q.rn <= t.id

runs for almost 30 seconds, while this one:

WITH t AS

(

SELECT 1 AS id

UNION ALL

SELECT 2

)

SELECT \*

FROM t

CROSS APPLY

(

SELECT TOP (t.id) m.\*

FROM master m

ORDER BY

id

) q

is instant.