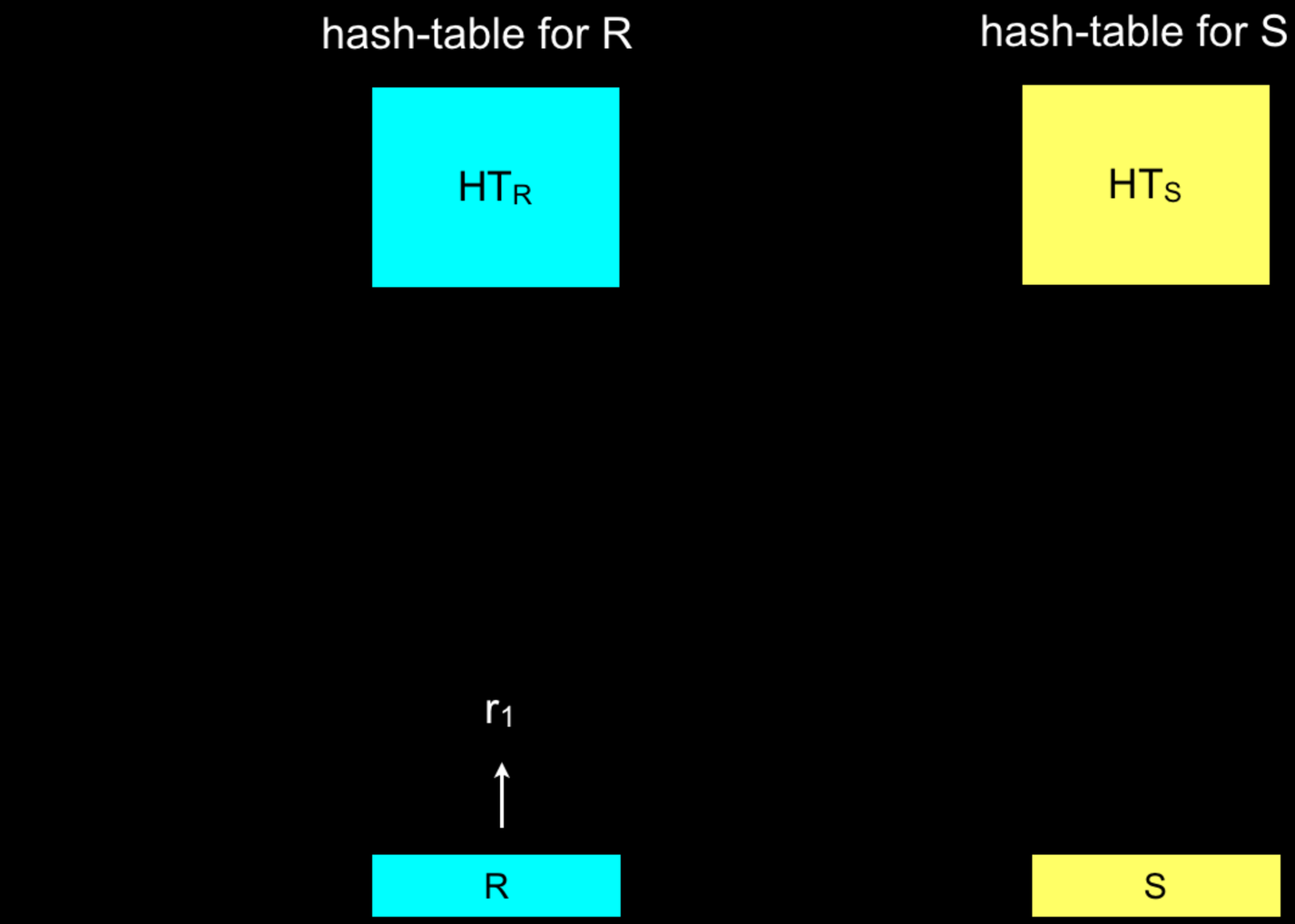
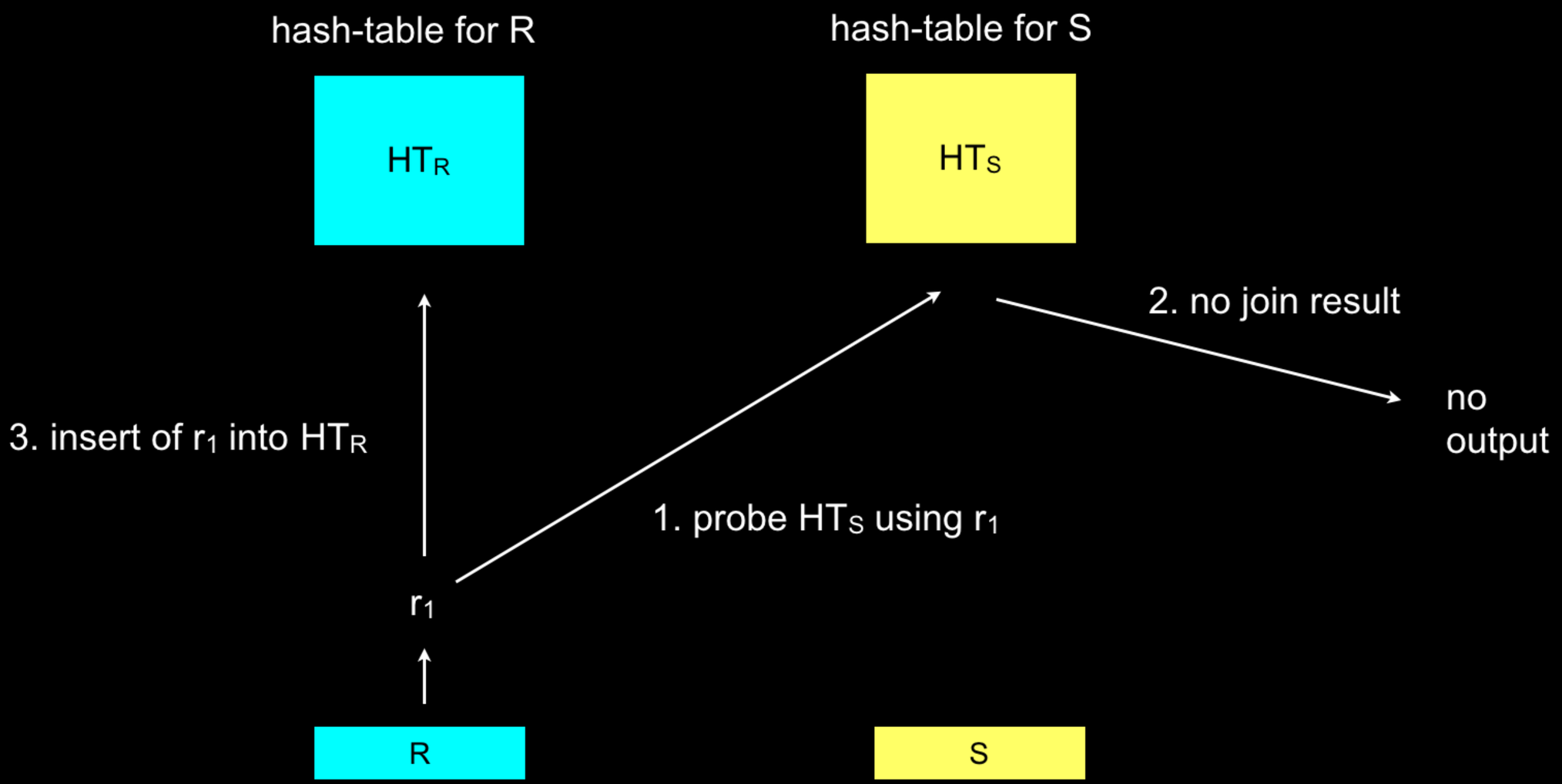
Example: Double-pipelined Hash Join

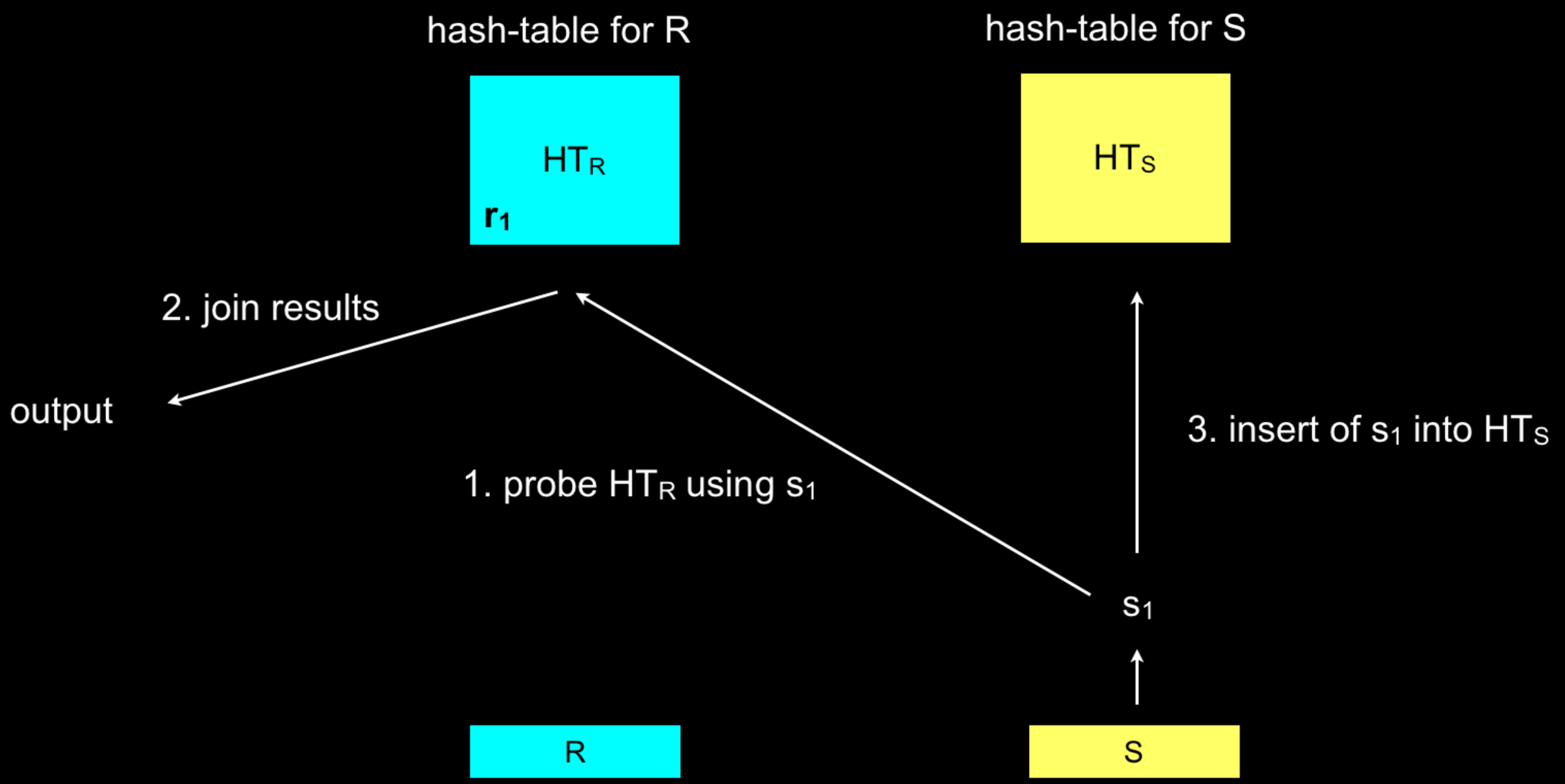


streaming sources

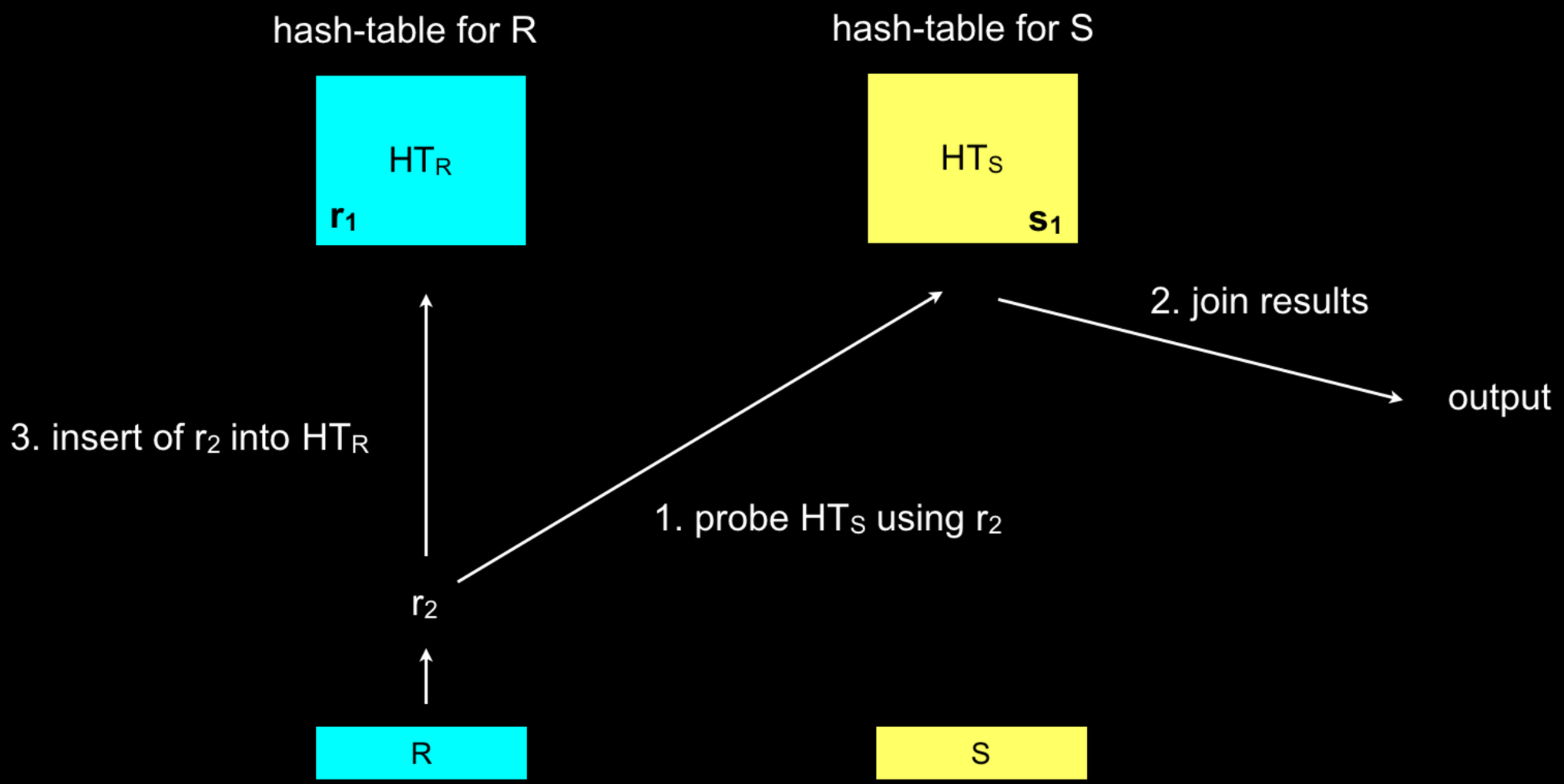
Example: Double-pipelined Hash Join



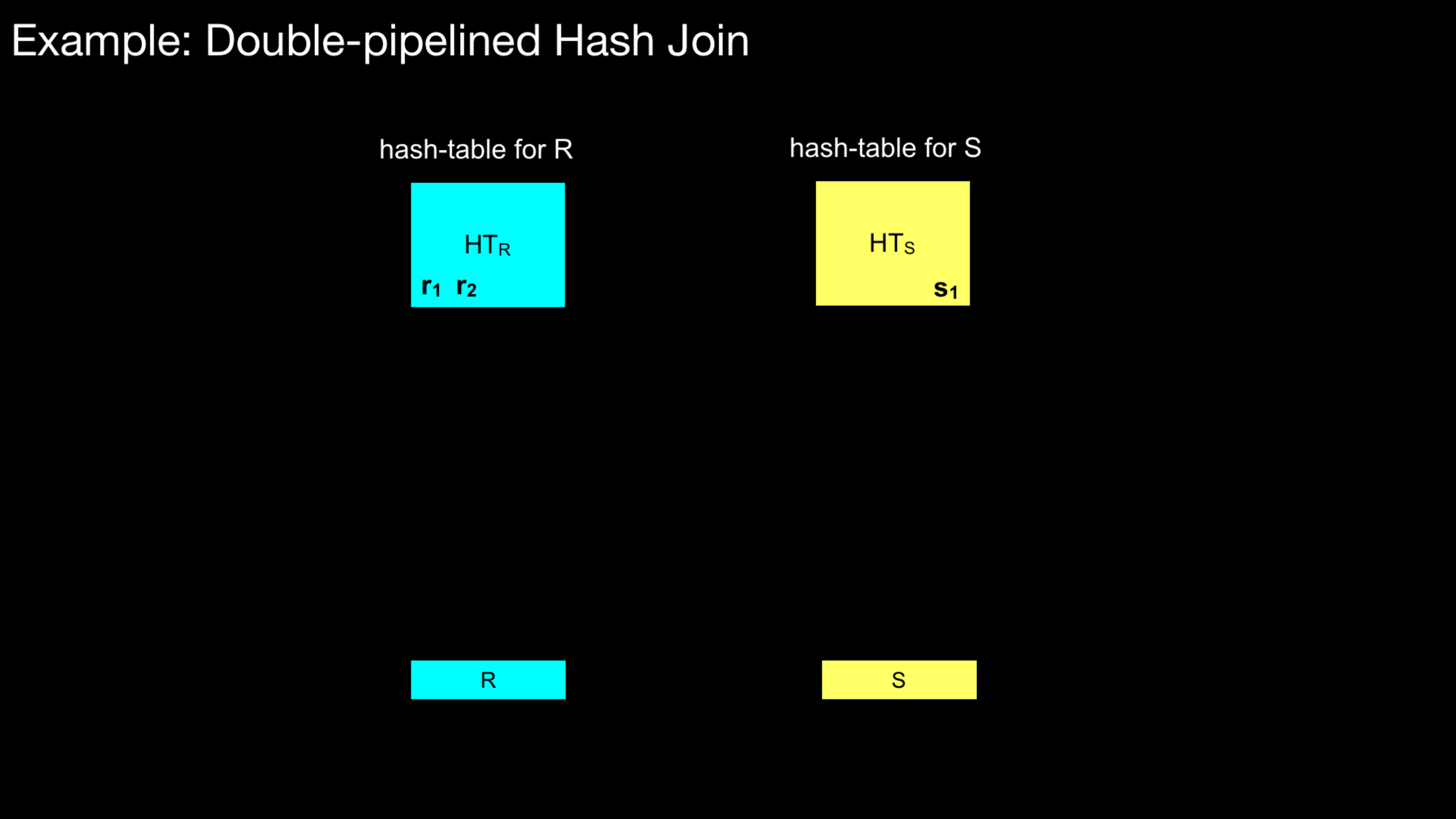
Example: Double-pipelined Hash Join



Example: Double-pipelined Hash Join



Example: Double-pipelined Hash Join



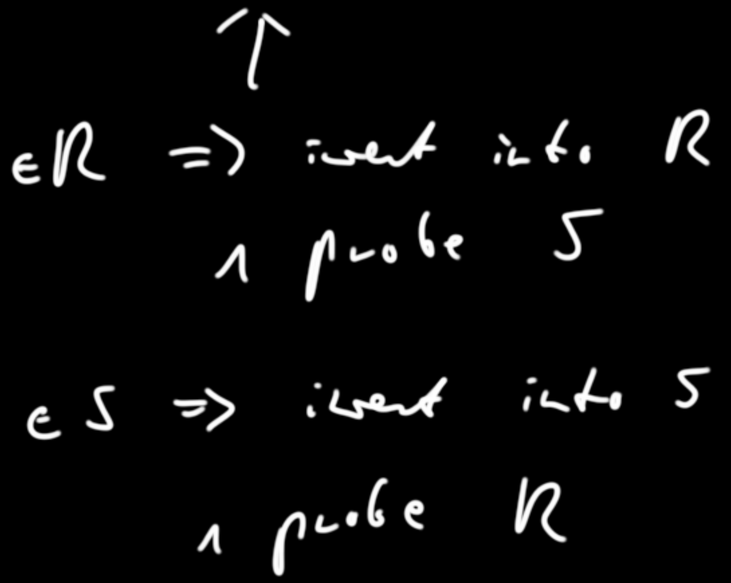
Double-pipelined Hash Join

R

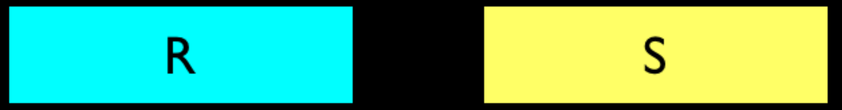
S

JP(r,s) := r.x == s.x //definition of the join predicate

probeAndInsert(tuple, indexToInsert, indexToProbe):

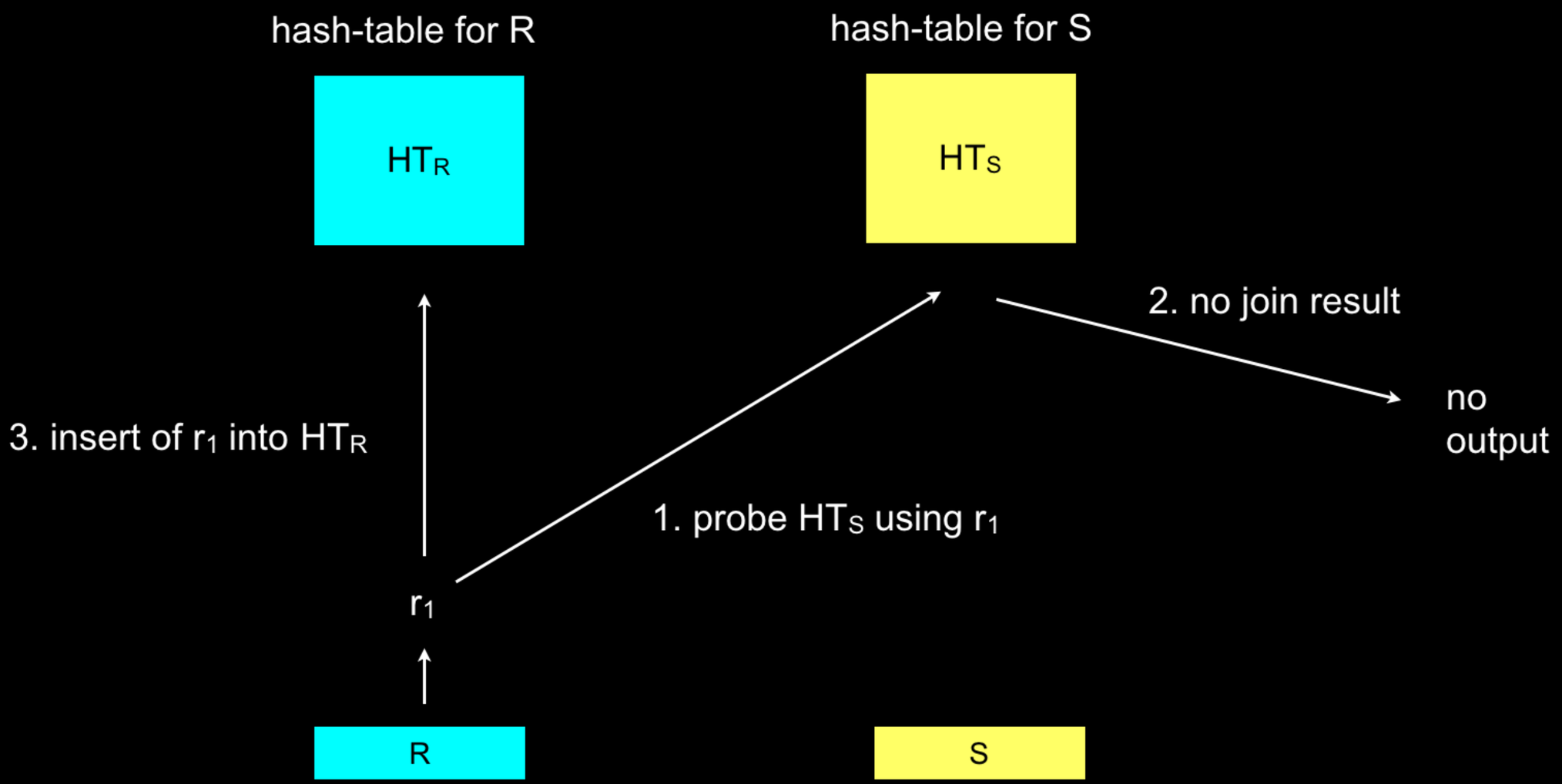


Double-pipelined Hash Join



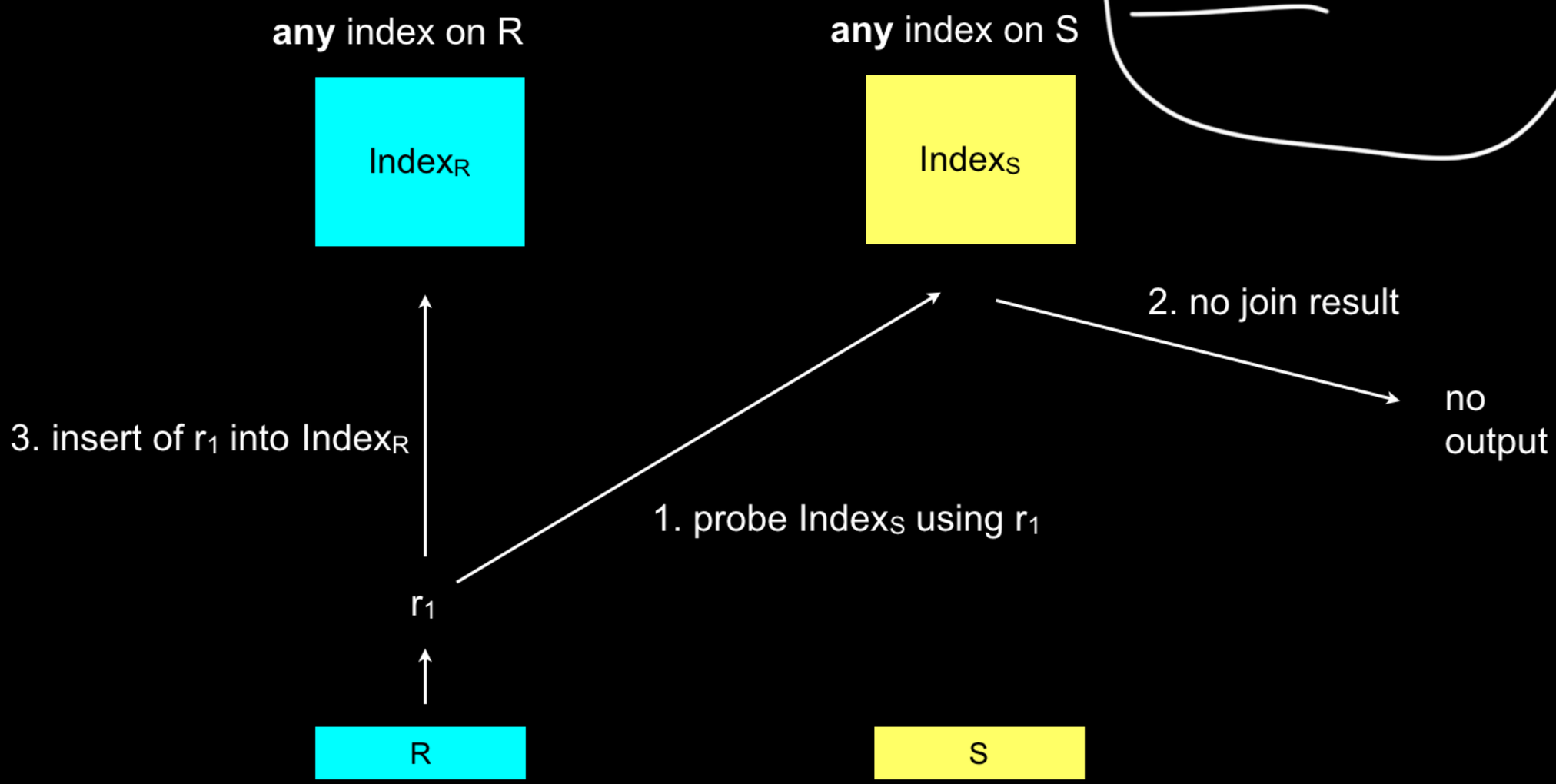

```
JP(r,s) := r.x == s.x                                          //definition of the join predicate
probeAndInsert( tuple,  indexToInsert, indexToProbe ):
        queryResultSet = indexToProbe.query( tuple.x );        //query index for this tuple.x (probe the index)
        If NOT queryResultSet.isEmpty():                       //did the query return results?
                joinResultSet = { tuple } × queryResultSet;    //cross product tuple with the query result
        indexToInsert.insert( tuple.x );                       //insert the tuple into the corresponding index
        return joinResultSet;                                  //return the actual join result for this tuple

DoublePipelinedHashJoin( R, S, JP(r,s) ):
        indexOnRX = new HashTable(); indexOnSX = new HashTable();   //initialize empty hash tables
        bool readFromR = TRUE;                                      //flag to determine relation to read from
        While R.hasNext() AND S.hasNext():                          //if both relations have items to read
                If readFromR:                                       //if we have to read from R in this round
                        output( probeAndInsert( R.next(), indexOnRX, indexOnSX ) );   //join the R tuple with the probed tuples of S
                Else:                                               //if we have to read from S in this round
                        output( probeAndInsert (S.next(), indexOnSX, indexOnRX ) );   //join the S tuple with the probed tuples of R
                readFromR = NOT readFromR;                          //switch to other relation for next round
        While R.hasNext():                                          //if only R has entries left
                output( probeAndInsert( R.next(), indexOnRX, indexOnSX ) );   //probe them against S and output
        While S.hasNext():                                          //if only S has entries left
                output( probeAndInsert( S.next(), indexOnSX, indexOnRX ) );   //probe them against R and output
```

Example: Double-pipelined Hash Join

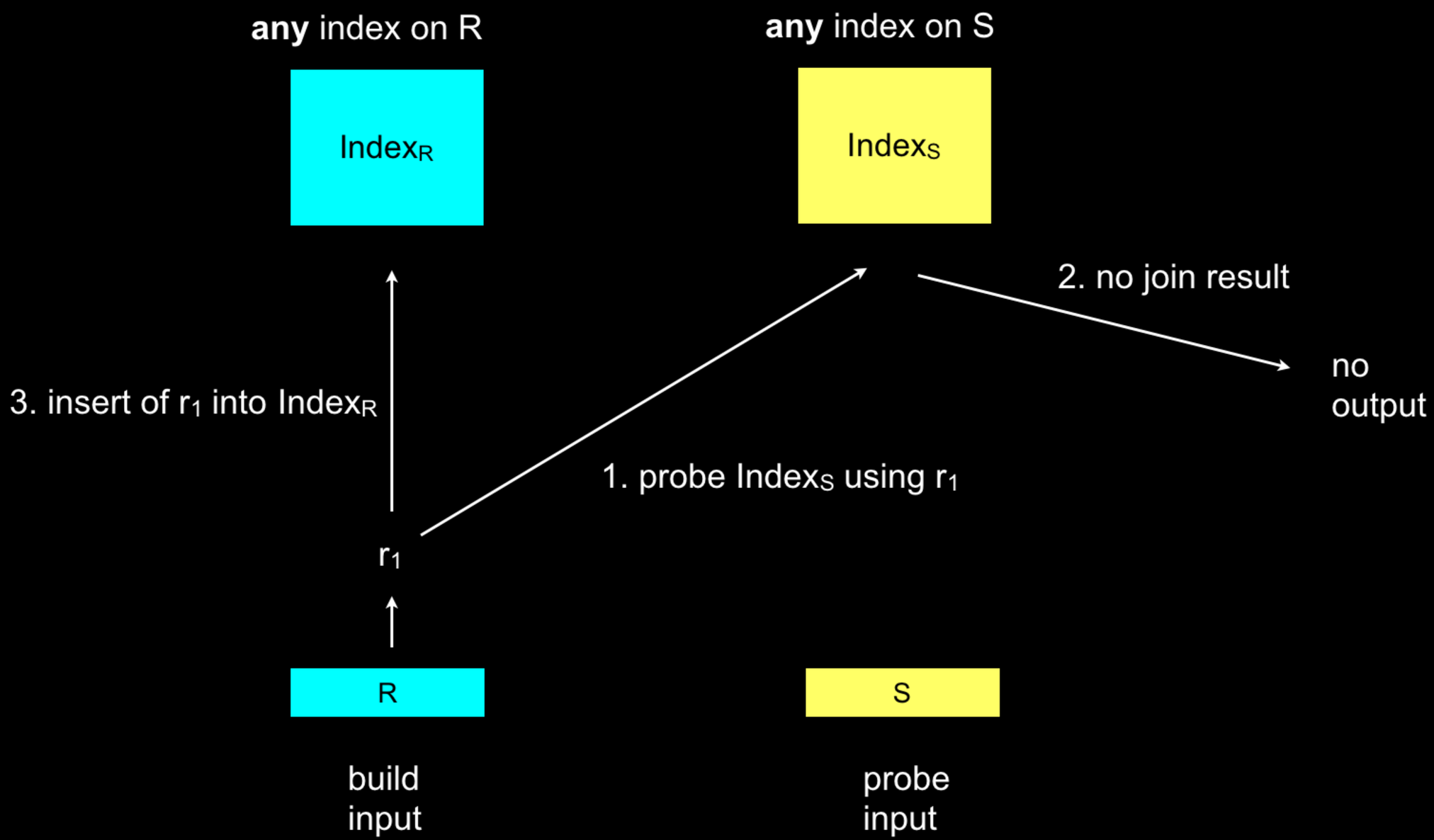


Example: Double-pipelined Index Join

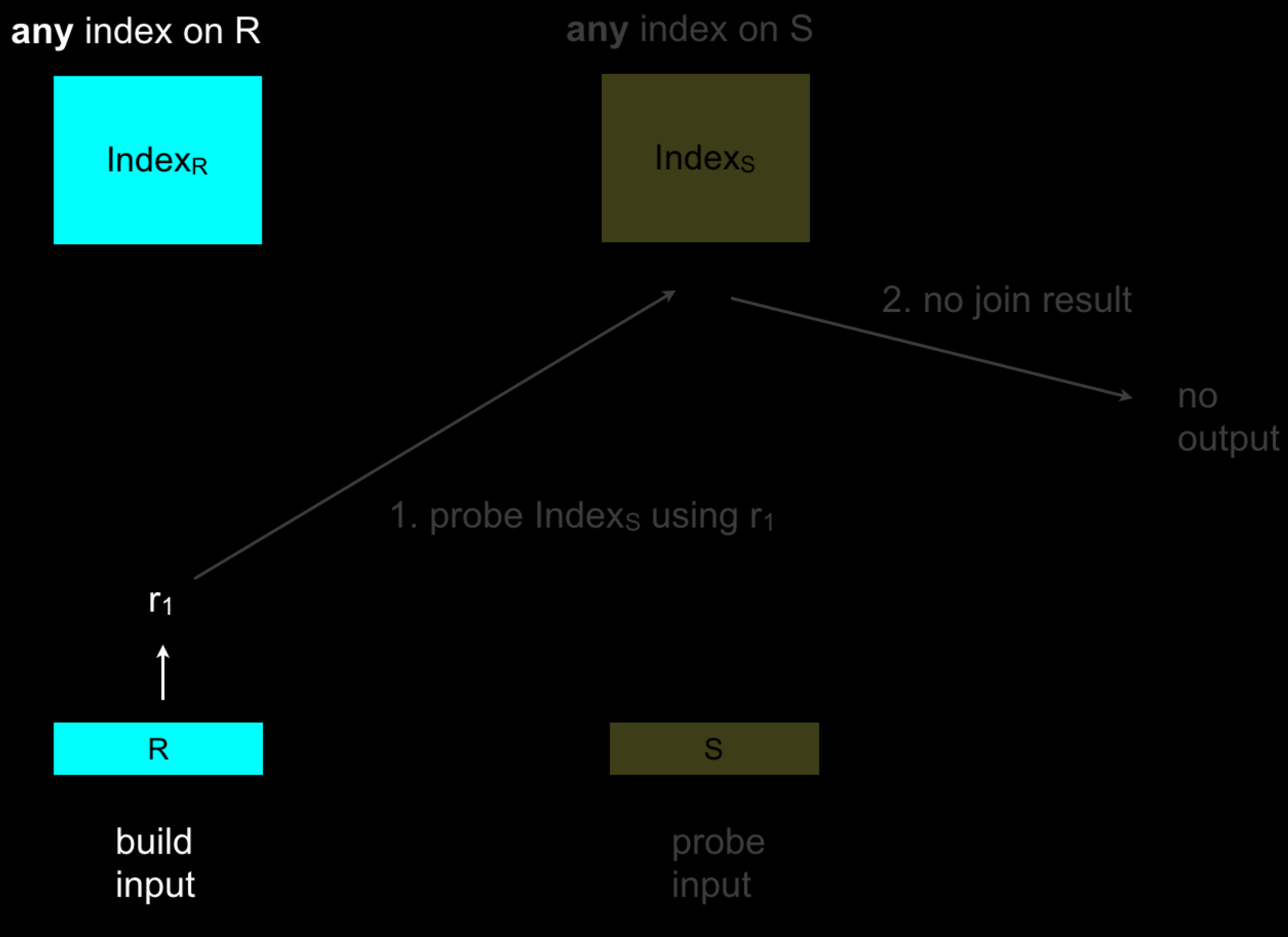
Relationship to INLJ?



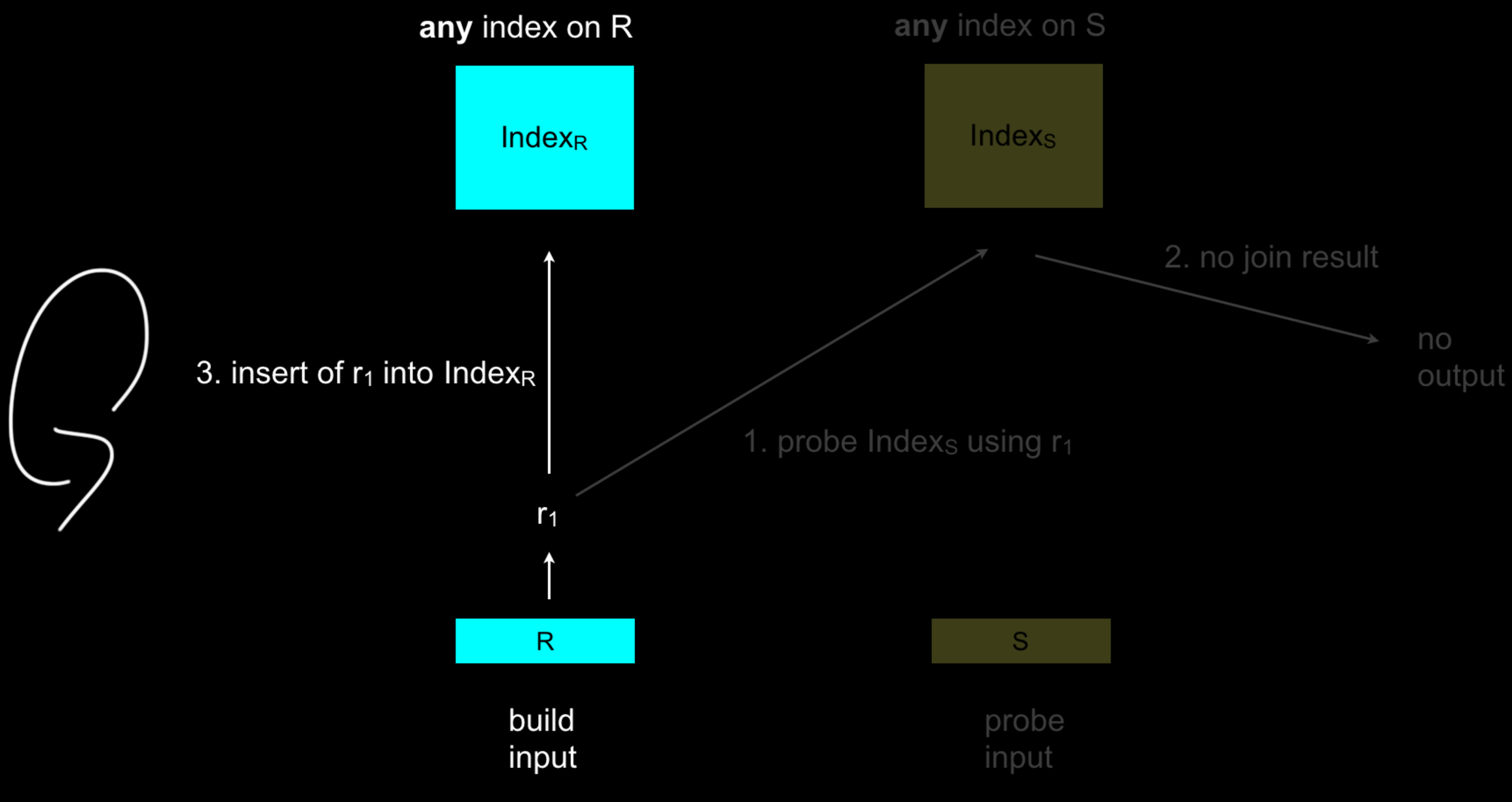
Example: Index Nested-Loop Join (Viewed Differently)



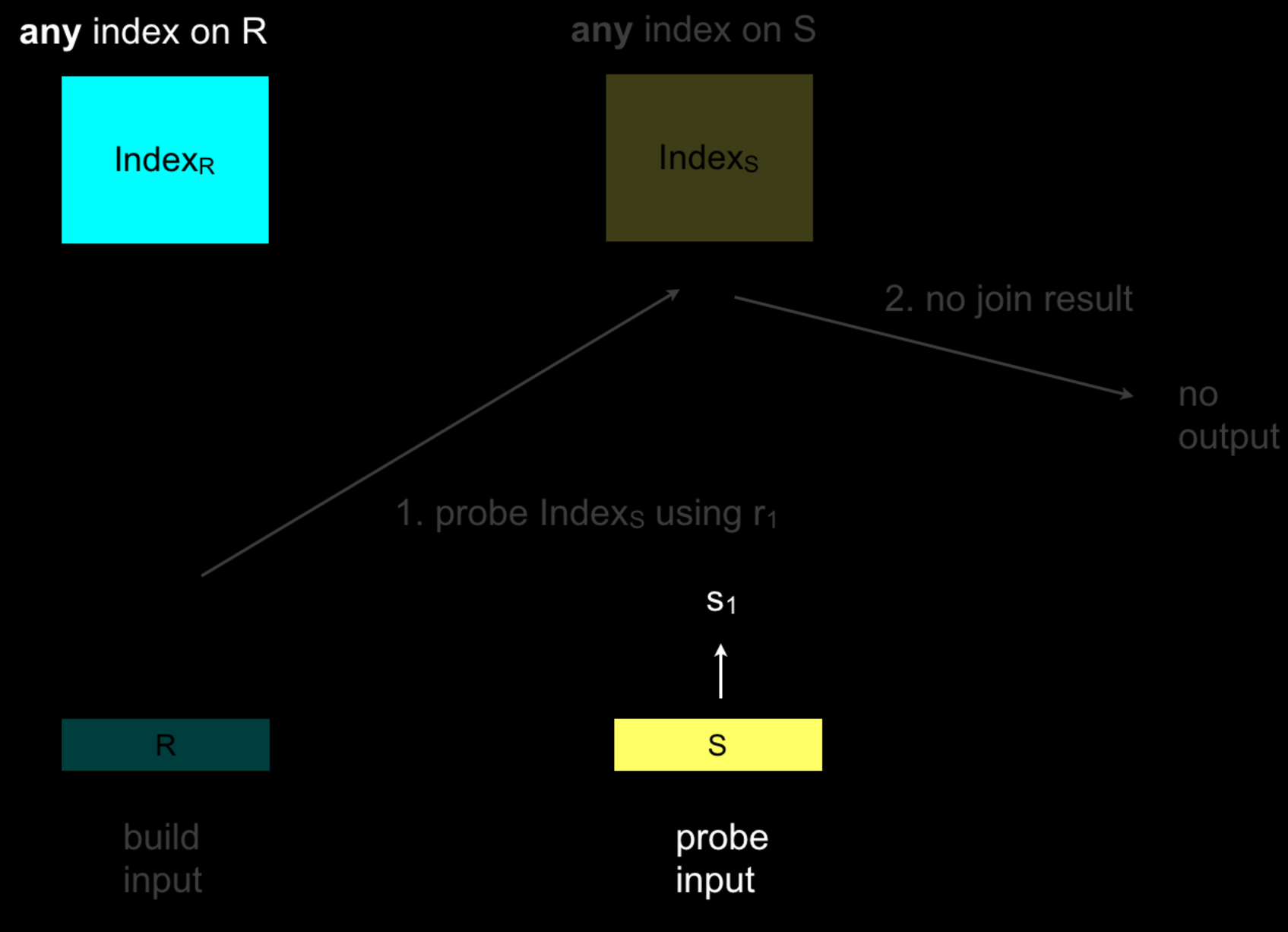
Build Phase of Index Nested-Loop Join



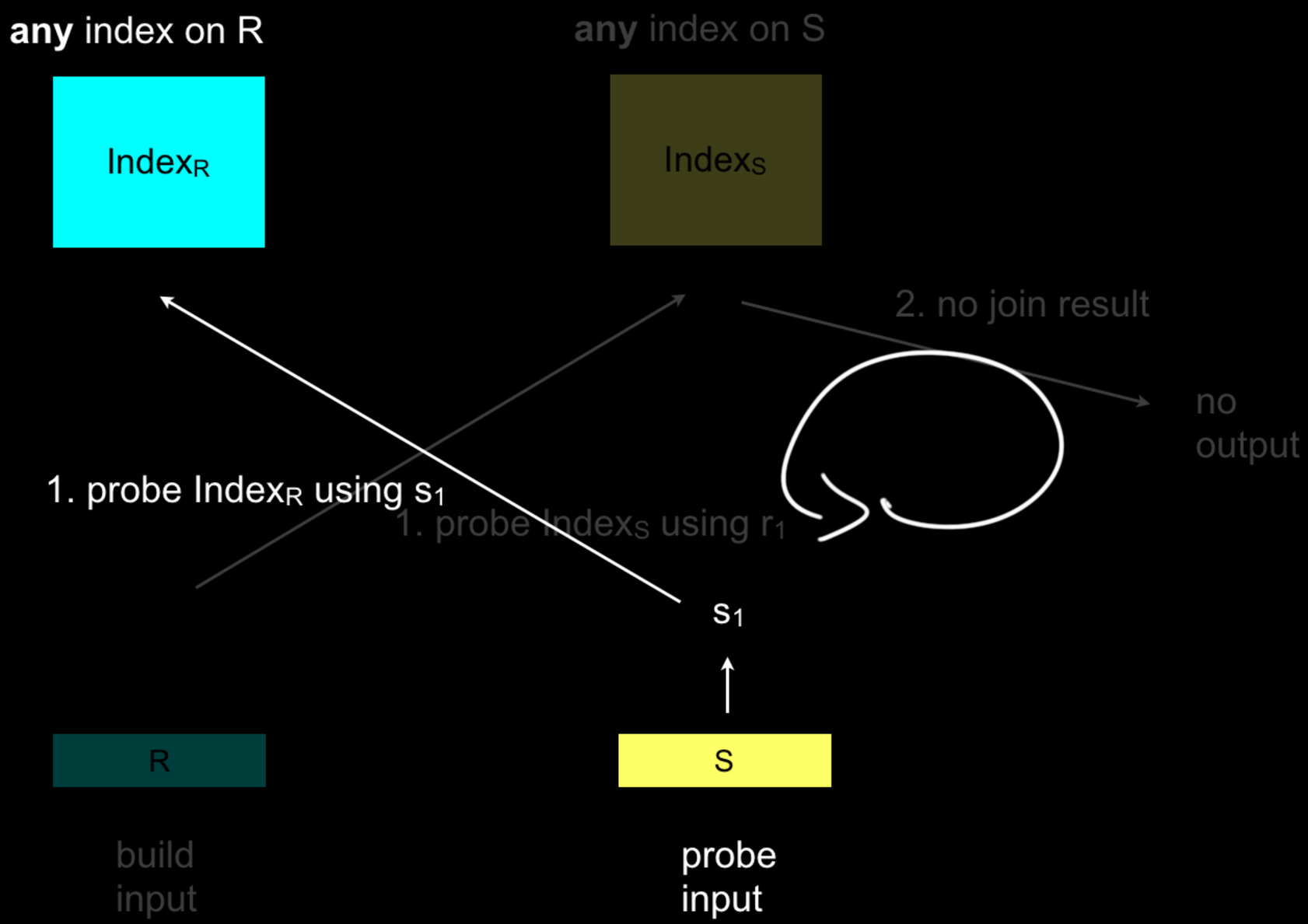
Build Phase of Index Nested-Loop Join



Probe Phase of Index Nested-Loop Join



Probe Phase of Index Nested-Loop Join

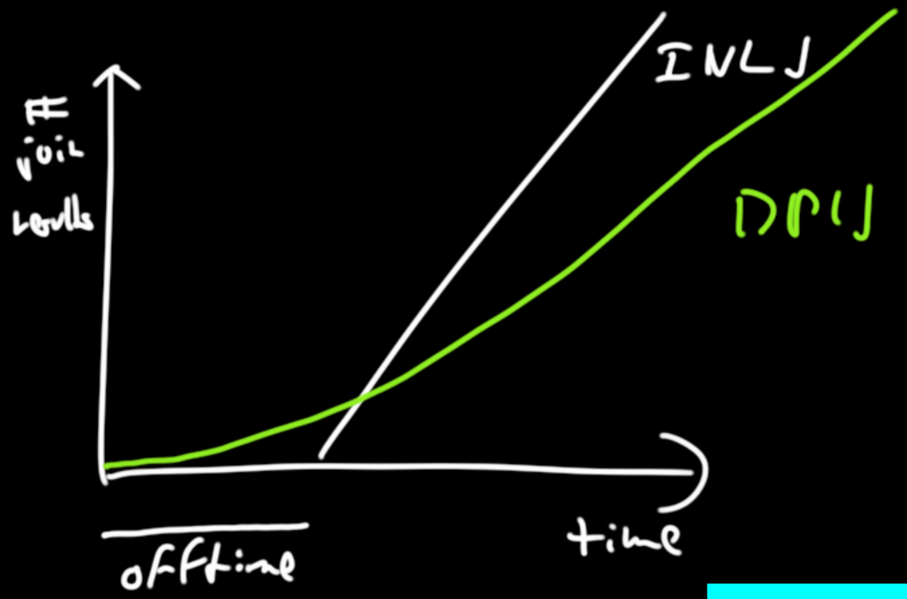


Difference of DPHJ over INLJ: Interleaving of Build and Probe Phases

any index on R

any index on S

DPIJ



Index$_R$

Index$_S$

R

S