

R JOIN S?

R

<i>c</i>	
40	T1
60	T2
30	T3
10	T4
20	T5

S

<i>c</i>	
10	T6
60	T7
40	T8
20	T9

Nested-Loop Join (NLJ)

For each $r \in R$ do

For each $s \in S$ do

if $r.C = s.C$ then output r,s pair

R	
40	T1
60	T2
30	T3
10	T4
20	T5

S	
10	T6
60	T7
40	T8
20	T9

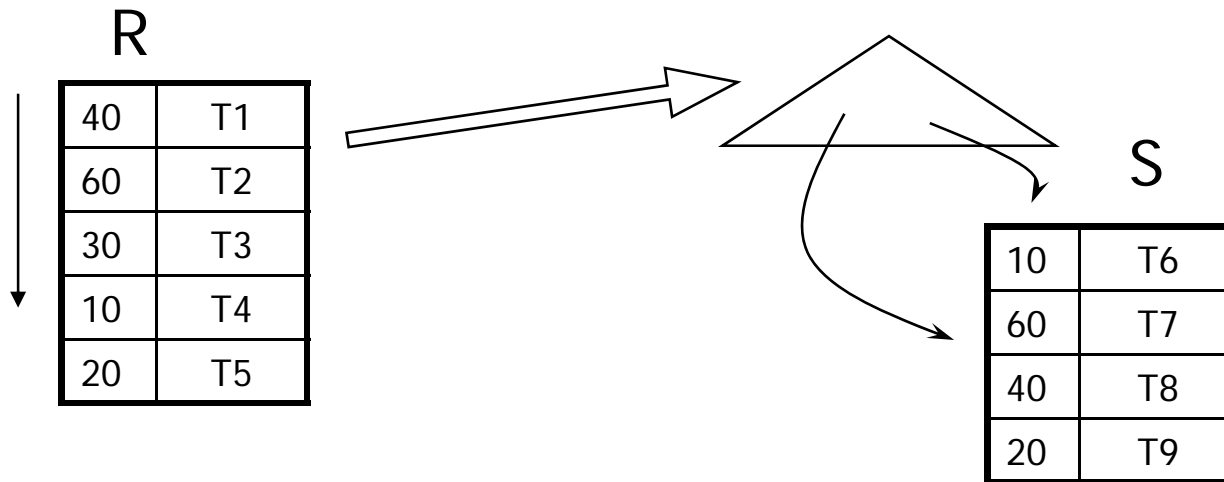
Index Join (IJ)

(1) Create an index for S.C if needed

(2) For each $r \in R$ do

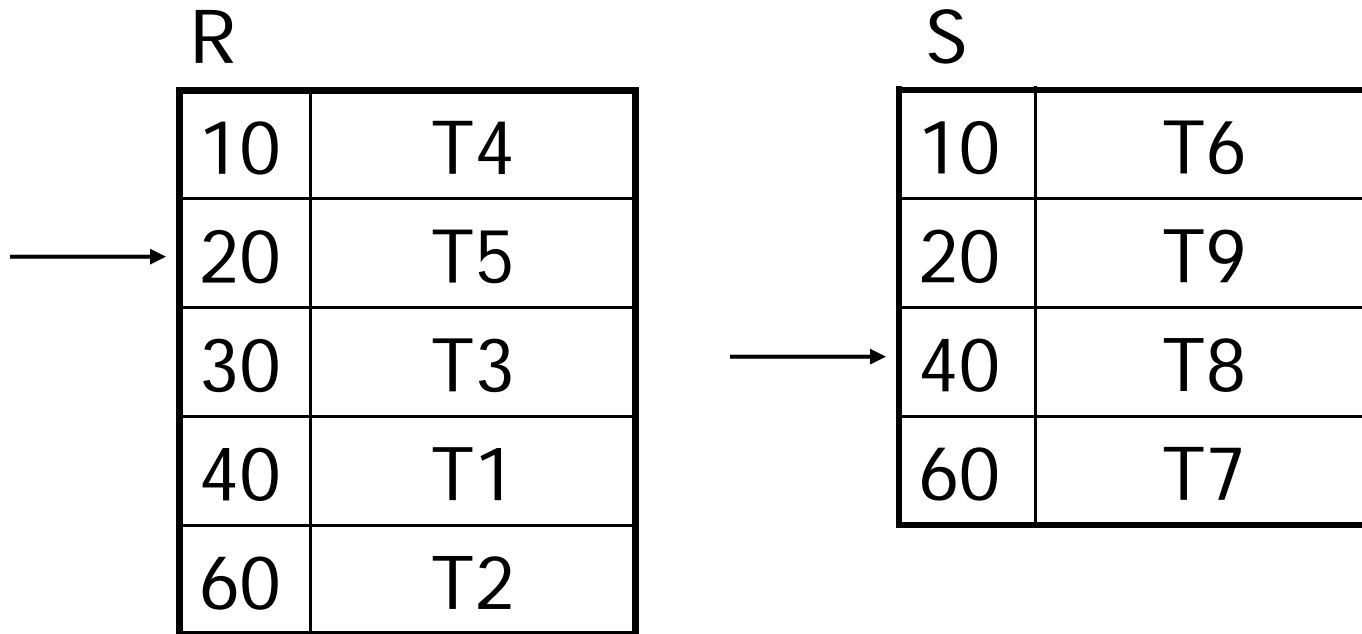
$X \leftarrow \text{index-lookup}(S.C, r.C)$

For each $s \in X$, output (r,s)



Sort-Merge Join (SMJ)

- Sort the relations first and join



Sort-Merge Join (SMJ)

(1) if R and S not sorted, sort them

(2) $i \leftarrow 1; j \leftarrow 1;$

While $(i \leq |R|) \wedge (j \leq |S|)$ do

 if $R[i].C = S[j].C$ then outputTuples

 else if $R[i].C > S[j].C$ then $j \leftarrow j+1$

 else if $R[i].C < S[j].C$ then $i \leftarrow i+1$

R

10	T4
20	T5
30	T3
40	T1
60	T2

S

10	T6
20	T9
40	T8
60	T7

Sort-Merge Join (SMJ)

Procedure outputTuples

While $(R[i].C = S[j].C) \wedge (i \leq |R|)$ do

$k \leftarrow j$;

 While $(R[i].C = S[k].C) \wedge (k \leq |S|)$ do

 output $R[i], S[k]$ pair;

$k \leftarrow k + 1$;

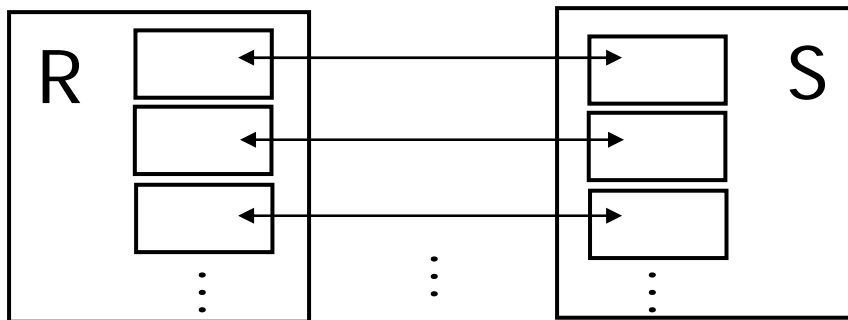
$i \leftarrow i + 1$;

Hash Join (HJ)

- Hash function $h(v)$, range $1 \rightarrow k$

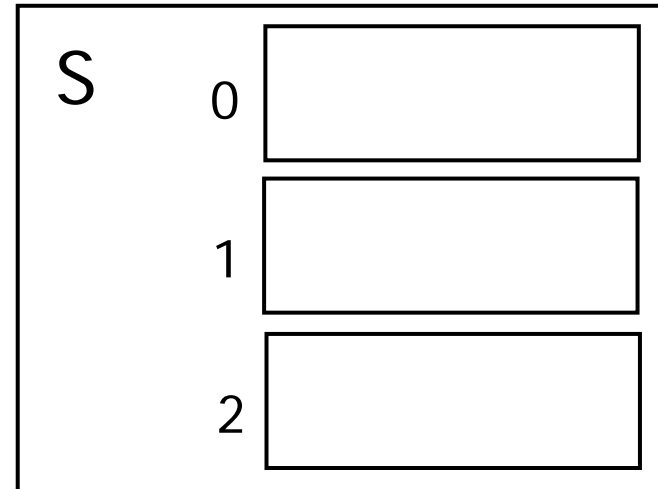
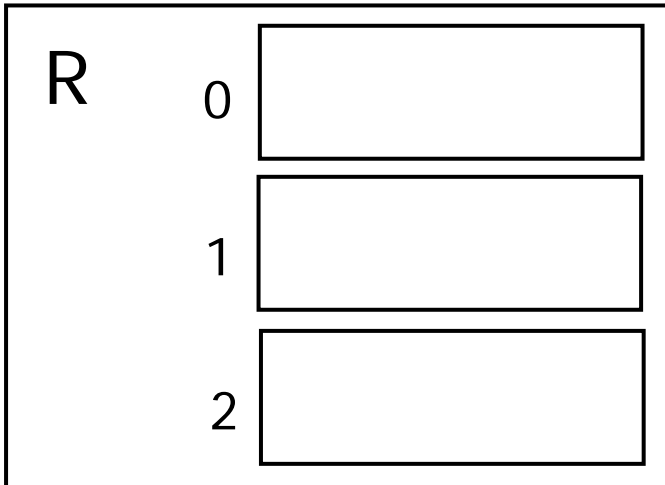
Algorithm

- (1) Hashing stage (bucketizing): hash tuples into buckets
 - Hash R tuples into G_1, \dots, G_k buckets
 - Hash S tuples into H_1, \dots, H_k buckets
- (2) Join stage: join tuples in matching buckets
 - For $i = 1$ to k do
 match tuples in G_i, H_i buckets



Hash Join (HJ)

- $H(k) = k \bmod 3$

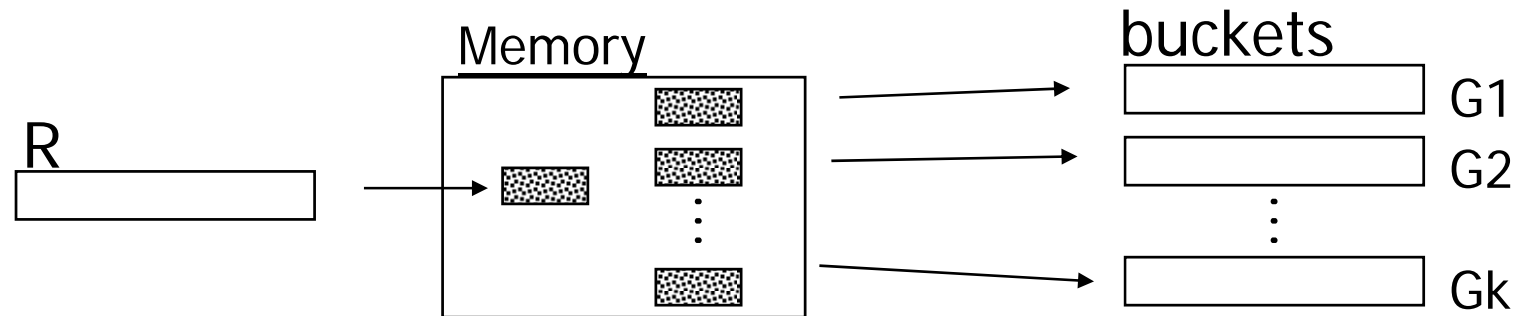


40	T1
60	T2
30	T3
10	T4
20	T5

10	T6
60	T7
40	T8
20	T9

Hash Join (HJ)

- Step (1): Hashing stage



- Step (2): Join stage

