```
proc $smjoin(R, S, `R_i = S_j')$

if $R$ not sorted on attribute $i$, sort it;
if $S$ not sorted on attribute $j$, sort it;

$Tr$ = first tuple in $R$;                                  // ranges over $R$
$Ts$ = first tuple in $S$;                                  // ranges over $S$
$Gs$ = first tuple in $S$;                    // start of current $S$-partition

while $Tr \neq eof$ and $Gs \neq eof$ do {

    while $Tr_i < Gs_j$ do
        $Tr$ = next tuple in $R$ after $Tr$;             // continue scan of $R$

    while $Tr_i > Gs_j$ do
        $Gs$ = next tuple in $S$ after $Gs$              // continue scan of $S$

    $Ts = Gs$;                                 // Needed in case $Tr_i \neq Gs_j$
    while $Tr_i == Gs_j$ do {                  // process current $R$ partition
        $Ts = Gs$;                                    // reset $S$ partition scan
        while $Ts_j == Tr_i$ do {                    // process current $R$ tuple
            add $\langle Tr, Ts \rangle$ to result;             // output joined tuples
            $Ts$ = next tuple in $S$ after $Ts$;} // advance $S$ partition scan
        $Tr$ = next tuple in $R$ after $Tr$;               // advance scan of $R$
        }                                    // done with current $R$ partition

    $Gs = Ts$;                       // initialize search for next $S$ partition

    }
```

Figure 14.8 Sort-Merge Join

sid	*sname*	*rating*	*age*
22	dustin	7	45.0
28	yuppy	9	35.0
31	lubber	8	55.5
36	lubber	6	36.0
44	guppy	5	35.0
58	rusty	10	35.0

Figure 14.9 An Instance of Sailors

sid	*bid*	*day*	*rname*
28	103	12/04/96	guppy
28	103	11/03/96	yuppy
31	101	10/10/96	dustin
31	102	10/12/96	lubber
31	101	10/11/96	lubber
58	103	11/12/96	dustin

Figure 14.10 An Instance of Reserves