2005 Systems Qual, Database Area: Part B

B.1 **[12 points; 2 for each part]** Consider two SQL tables $R(A, B)$ and $S(A, B)$. You want to perform the SQL intersection (`intersect`, `intersect-all`), union (`union`, `union-all`), and difference (`except`, `except-all`) operations on these tables, but unfortunately you only have a simplified version of SQL: `select-from-where` queries, with subqueries permitted in the `where` clause only. For each of the following problems, either write the simplified SQL query to perform the operation, or write "cannot be expressed" (no justification needed). Queries will be graded on simplicity as well as correctness.

(a)	R `intersect` S	A is a key for R and B is a key for S
(b)	R `intersect-all` S	Neither R nor S has a key
(c)	R `union` S	A is a key for R and B is a key for S
(d)	R `union-all` S	Neither R nor S has a key
(e)	R `except` S	A is a key for R and B is a key for S
(f)	R `except-all` S	Neither R nor S has a key

B.2 **[8 points]** You are performing query optimization for the join of two tables, R and S. Your statistics on the cardinalities (sizes) of R and S are outdated. Your physical query operators are implemented using iterators. Suppose as part of plan costing, you are estimating the total number of `GetNext` calls during query exeuction for different plans. Consider the following three plans:

1. (`scan`(R)) `nested-loops-join` (`scan`(S))
2. (`scan`(R)) `hash-join` (`scan`(S))
3. (`sort`(`scan`(R)) `merge-join` (`sort`(`scan`(S))

Compare the three plans in terms of how sensitive the accuracy of your #`GetNext` estimates are to erroneous cardinality statistics for R and S. Specifically, order the plans from least sensitive to most sensitive, and briefly justify your ordering.