Sybase hints

The following table highlights SQL hints based on Sybase hints optimization.

Category	Hint	Notes
Logical	distinct	No explicit implementation.
Logical	group	No explicit implementation.
Logical	g_join	No explicit implementation.
Logical	nl_g_join	Not applicable for: statements with chained queries; select statements with group by clause and having clause or group by clause and order by clause.
Logical	m_g_join	Not applicable for: statements with chained queries; select statements with group by clause and having clause or group by clause and order by clause.
Logical	join	No explicit implementation.
Logical	nl_join	Not applicable for: select statements with group by clause and having clause or group by clause and order by clause.
Logical	m_join	Not applicable for: select statements with group by clause and having clause or group by clause and order by clause.
Logical	h_join	Not applicable for: select statements with group by clause and having clause or group by clause and order by clause.
Logical	union	No explicit implementation.
Logical	scan	No explicit implementation.
Logical	scalar_agg	Only used in combination with other operators. It does not change the execution plan itself.
Logical	sequence	Is a keyword that will be used in the implementation of scalar_agg operator.
Logical	hints	We don't support a combination of hints.
Logical	prop	Uses a set of pre-defined values.
Logical	table	Used only in combination with other operators, when referring tables from subqueries.
Logical	work_t	This operator is applicable only together with store operator.
Logical	in	Used only in combination with other operators, when referring tables from subqueries.
Logical	subq	Used only in combination with other operators, when referring tables from subqueries.
Physical	distinct_sorted	Only for SELECT statements containing DISTINCT, and only for tables.
Physical	distinct_sorting	Only for SELECT statements containing DISTINCT, and only for tables.
Physical	distinct_hashing	Only for SELECT statements containing DISTINCT, and only for tables.
Physical	group_sorted	Only for SELECT statements (not working for views) with no having and no order by clause.
Physical	group_hashing	Only for SELECT statements (not working for views) with no having and no order by clause.
Physical	group_interting	Not implemented.
Physical	append_union_all	Not applicable for: UNION chained clauses, nested sub-selects in a from clause, if a group by clause is present or if scalar aggregation is present.
Physical	merge_union_all	Not applicable for: UNION ALL chained clauses, nested sub-selects in a from clause, or if a group by clause is present.
Physical	merge_union_distinct	Not applicable for: UNION ALL chained clauses, nested sub-selects in a from clause, or if a group by clause is present.
Physical	hash_union_distinct	Not applicable for: UNION ALL chained clauses, nested sub-selects in a from clause, if a group by clause is present, or if scalar aggregation is present.
Physical	i_scan	Applied to all table references in the from clause of the main select and of the sub select statements except: 1. statement has sub-selects. 2. table references has no indexes.

Physical	t_scan	Applied to all the table references in the from clause of the main select and of the sub select statements except:

		On Sybase 12.5 not applied for tables in the main query if:
		
		1. statement has chained queries.
		2. Sub queries have group by and having clauses; and not applied to the tables in sub selects if:
		    a. has select statements in from clause of the main select.
		    b. sub queries have group by and having clauses.
		    c. statement has select statements in select clause.
		    d. statement has parent statement and insert statement;
		
		On Sybase 15 not applied for tables in sub selects if:
		
		1. has select statements in from clause of the main select.
		2. statement has chained queries.

Physical	m_scan	Applied for all tables if in the where clause there is a condition like:
		
		    table1.indexedColumn1 condition body OR
		
		    table1.indexedColumn2 condition body;
		
		Not applied if the LIKE operator is used. For columns that belong to a primary key only the first column is considered.

Physical	store	--
Physical	store_index	--
Physical	sort	--
Physical	xchg	--