Create a Sailors Table about there age and name rating :-

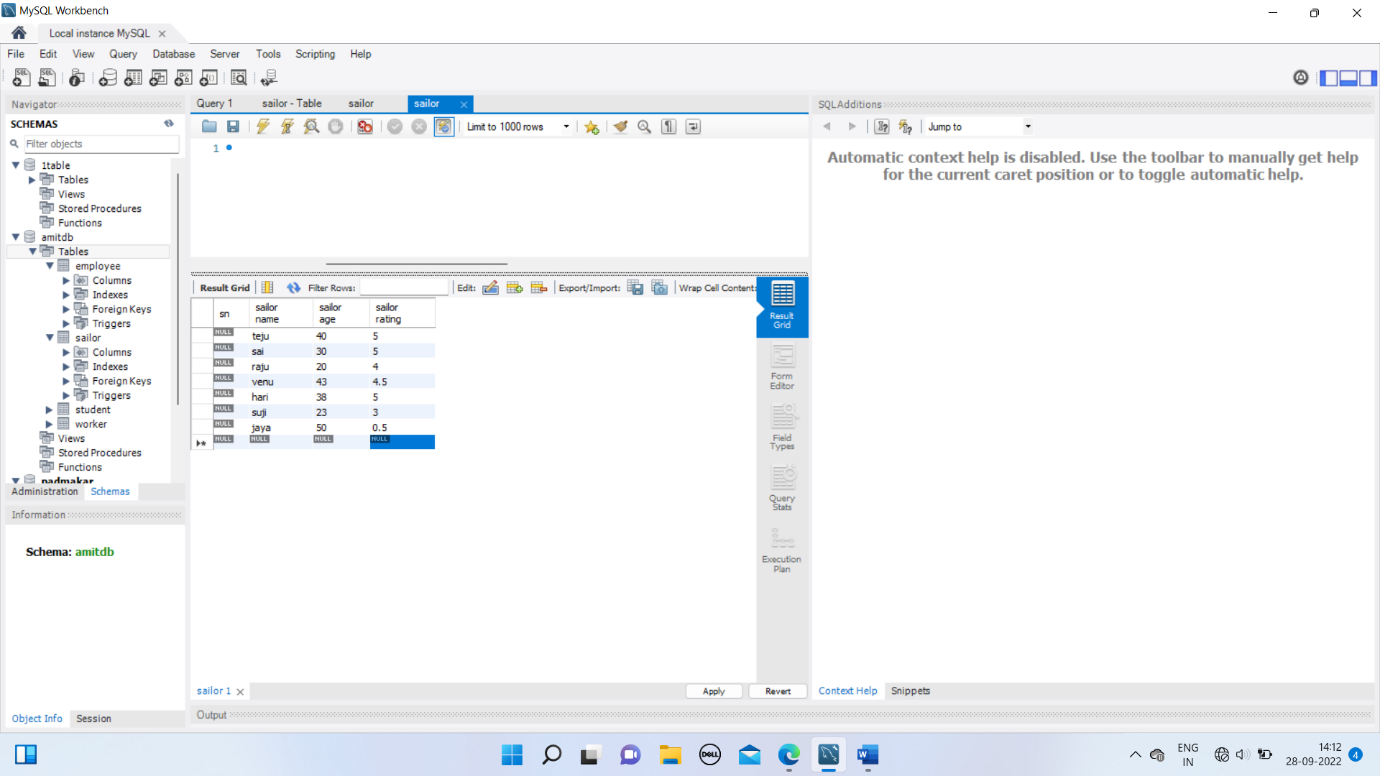
## SELECT[ DISTINCT ] select-list FROM from-list WHERE qualification Such a query intuitively corresponds to a relational algebra expression involving selections, projections, and cross-products. Every query must have a SELECT clause, which specifies columns to be retained in the result, and a FROM clause, which specifies a cross-product of tables. The optional WHERE clause specifies selection conditions on the tables mentioned in the FROM clause. Let us consider a simple query.

|  |  |  |
| --- | --- | --- |
| teja | 50 | rating |
| sai | 40 | 5 |
| raju | 30 | 4 |
| venu | 20 | 4.5 |
| hari | 43 | 5 |
| suji | 38 | 3 |
| jaya | 23 | 0.5 |

Find all sailors with a rating above 3:-

• The DISTINCT keyword is optional. It indicates that the table computed as an answer to this query should not contain duplicates, that is, two copies of the same row. The default is that duplicates are not eliminated. the syntax of a basic SQL query, they don’t tell us the meaning of a query. The answer to a query is itself a relation which is a multiset of rows in SQL whose contents can be understood by considering the following conceptual evaluation strategy: 1. Compute the cross-product of the tables in the from-list. 2. Delete those rows in the cross-product that fail the qualification conditions. 3. Delete all columns that do not appear in the select-list. 4. If DISTINCT is specified, eliminate duplicate rows.

|  |  |  |
| --- | --- | --- |
| sname | age | rating |
| teja | 50 | 4 |
| sai | 40 | 5 |
| raju | 30 | 4 |
| venu | 20 | 4.5 |
| hari | 43 | 5 |

******