

rd(TS, Pattern)

register(memactor, spawn(?MODULE, db, []))

esame:rd(TS, Pattern) -> memactor ! {out, TS, Pattern, self()}

Pid ! {error}

ets:info(Name)

Pid ! {error}

```
PatternList = tuple_to_list(Pattern),
MyFun = fun(Value, Acc) ->
    ValueList = tuple_to_list(Value),
    case db:match(PatternList, ValueList) of
        true ->
            [Value | Acc];
        _ ->
            Acc
    end
Trovato = ets:foldl(MyFun, [], TS),
```

Pid ! {error}

case Trovato of

[] ->
Pid ! {[]}

_ ->
Pid ! {ok, Trovato}

end

Pid ! {ok, Trovato}