rd(TS, Pattern, Timeout)

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}

```
T = Timeout - 100,
timer:sleep(100),
case Trovato of
    [] ->
        memactor ! {rd, TS, Pattern, Pid, T};
    _ ->
        Trovato
        Pid ! {ok, Trovato}
end
```

Pid ! {ok, Trovato}