```clojure
(defn pattern-left-join
  [graph partial-result pattern]
  (let [cols (:cols (meta partial-result))
        total-cols (calc-new-columns cols pattern)
        pattern->left (matching-vars pattern cols)]

    ;; iterate over partial-result, lookup pattern
    (with-meta
      (for [left-row partial-result
                 ;; convert bindings in left-row into the
                 ;; pattern to lookup in the graph
            :let [lookup (modify-pattern left-row
                                          pattern->left
                                          pattern)]
            right-row (gr/resolve-pattern graph lookup)]
        (concat left-row right-row))
      {:cols total-cols})))
```

Other Considerations

- Join Ordering (optimizing)
- Operations
  - disjunctions
  - bindings
  - subtractions
  - aggregates
- Durability
  Indexes in:
  - B*-tree
  - Hitchhiker trees
  - etc.

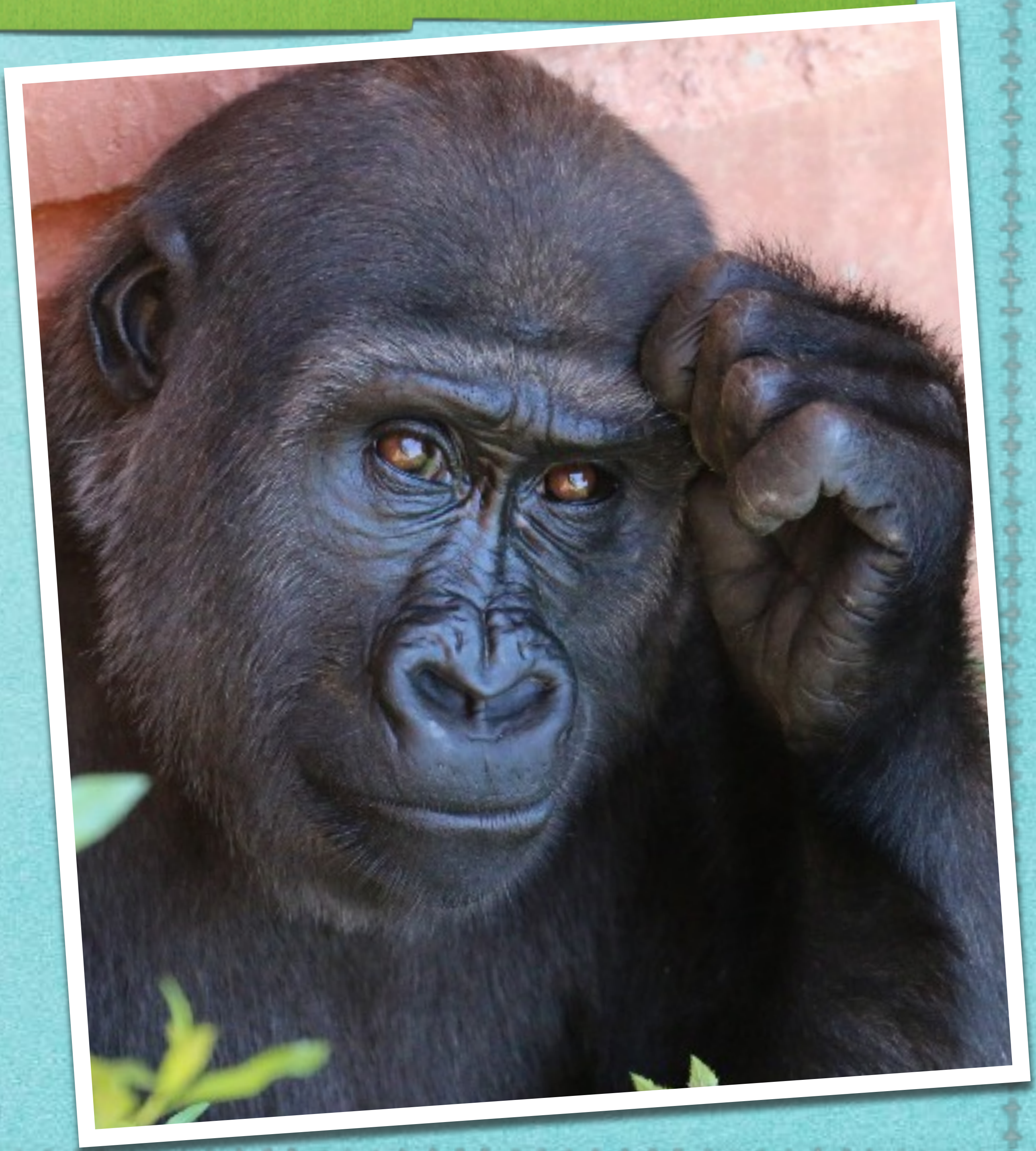

Photo: Rob Schreckhis @robschreckhise