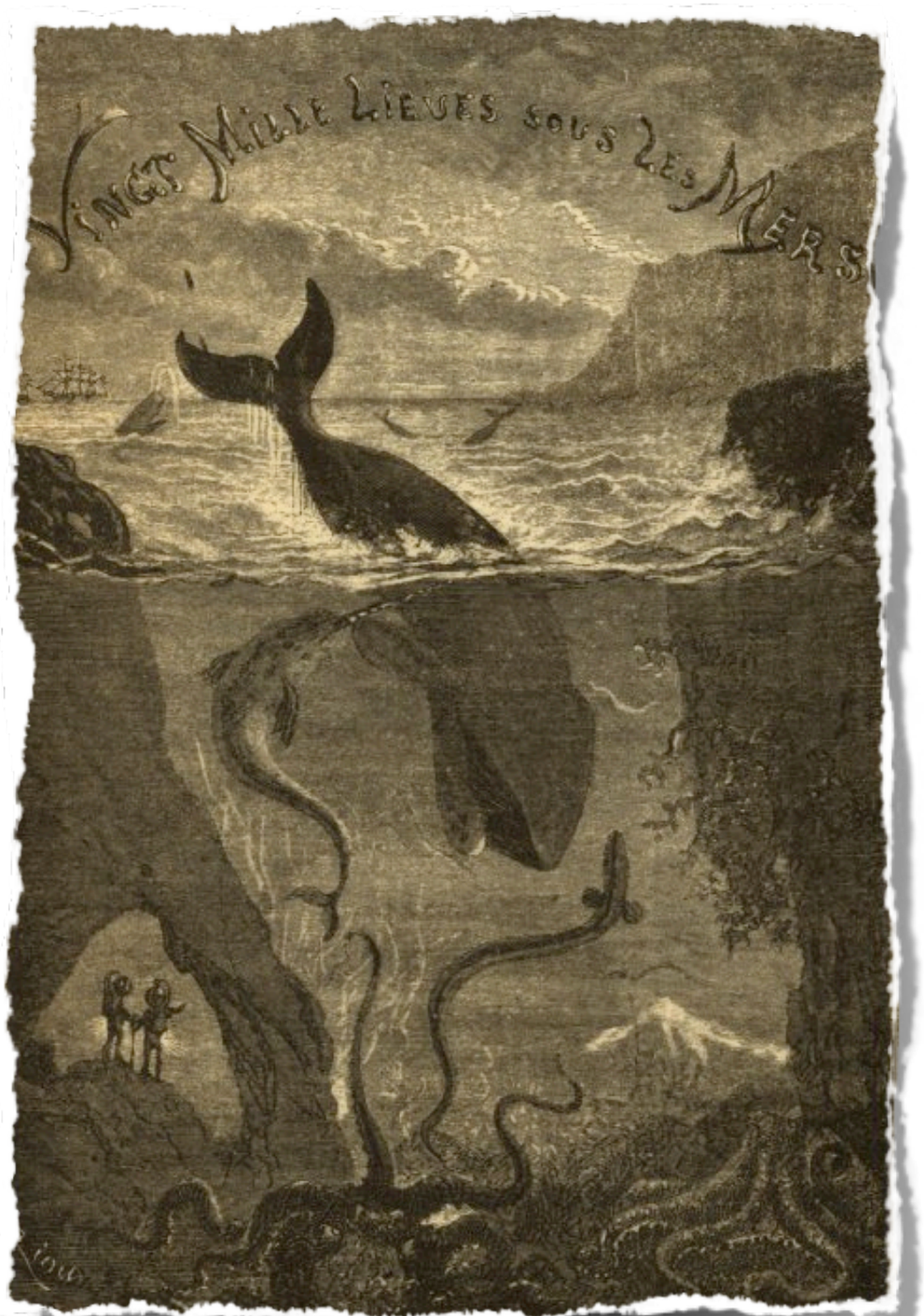
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Transformation Lifecycle Management with Nautilus

Melanie Herschel, Torsten Grust, and Tim Belhomme
University of Tübingen
Germany

Workshop on Quality in Databases 2011
collocated with VLDB

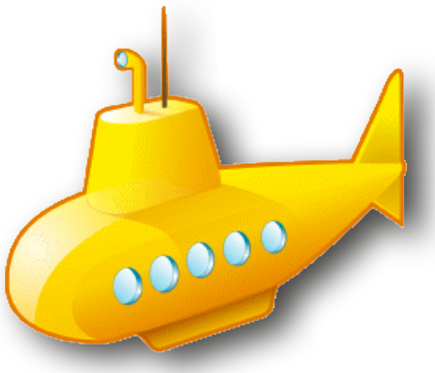
What is Nautilus?



"The **deepest parts of the ocean** are totally unknown to us[...] **What goes on** in those distant depths? What creatures **inhabit**, or could inhabit, those regions twelve or fifteen miles beneath the surface of the water? It's almost beyond **conjecture**" Jules Verne, 20.000 Leagues under the Sea, Chapter 2.

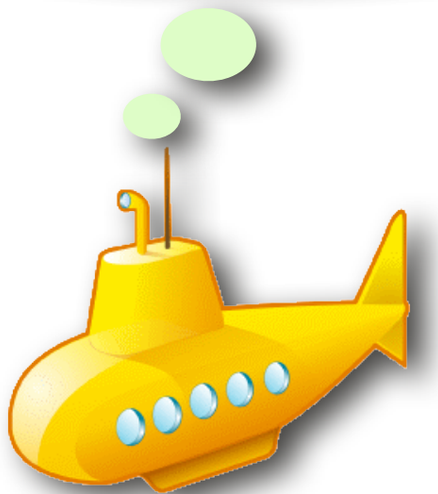
What is Nautilus?

"The **deepest parts of the ocean** are totally unknown to us[...] **What goes on** in those distant depths? What creatures **inhabit**, or could inhabit, those regions twelve or fifteen miles beneath the surface of the water? It's almost beyond **conjecture**" Jules Verne, 20.000 Leagues under the Sea, Chapter 2.



What is Nautilus?

"The **deepest parts of the ocean** are totally unknown to us[...] **What goes on** in those distant depths? What creatures **inhabit**, or could inhabit, those regions twelve or fifteen miles beneath the surface of the water? It's almost beyond **conjecture**" Jules Verne, 20.000 Leagues under the Sea, Chapter 2.



What is Nautilus?

What happens within transformation?

What data?

How is data combined?

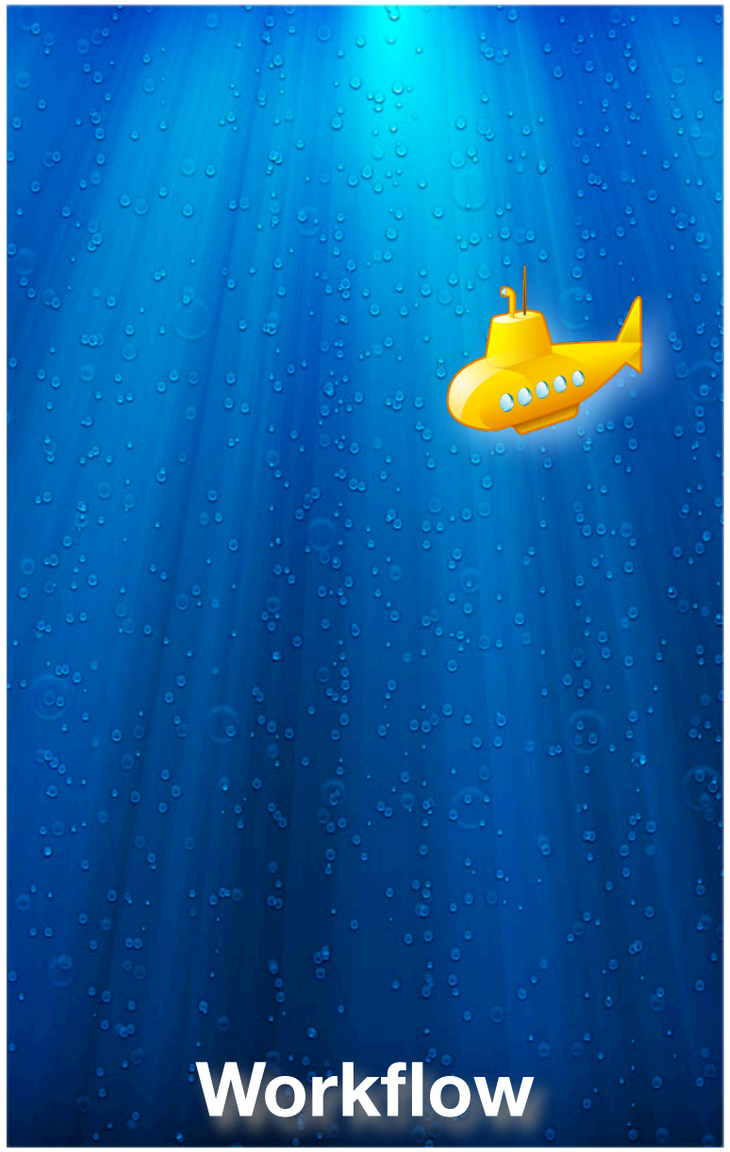
Developing data transformations - state of the art

- Manual trial-and-error process
- No systematic tool exists that supports the complete Analyze-Fix-Test (AFT) cycle.
- New requirements lead to further cycles.

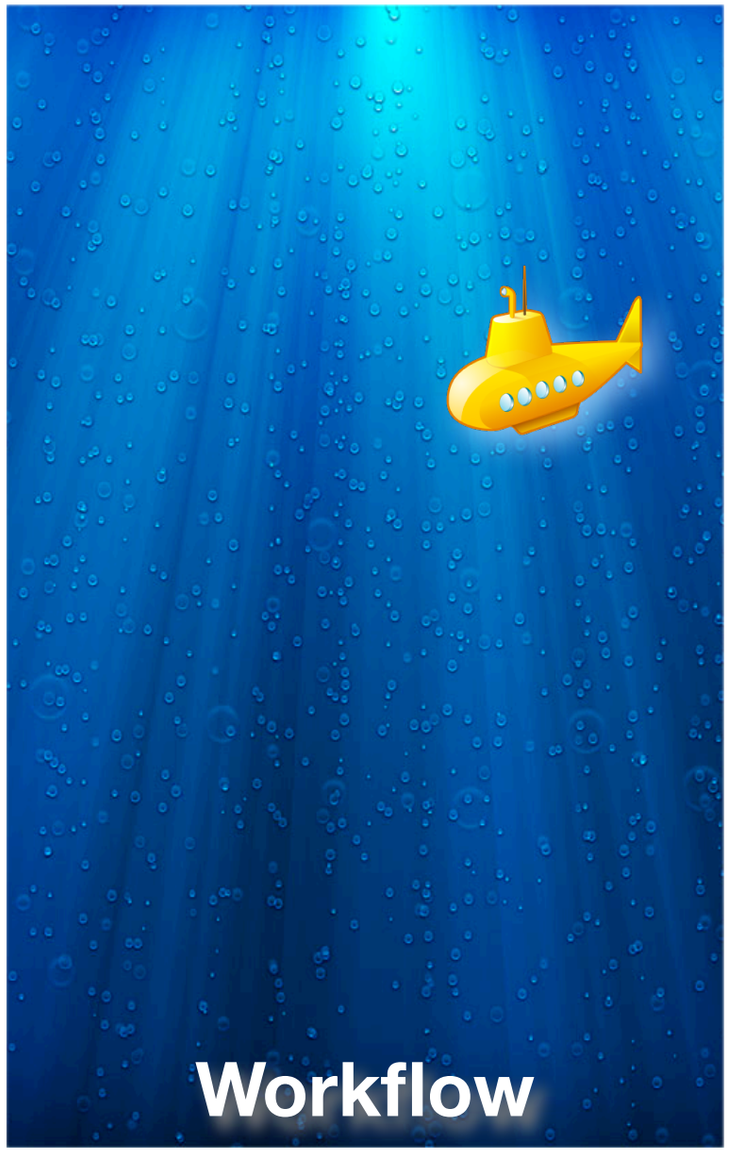
Why Transformation Lifecycle Management?

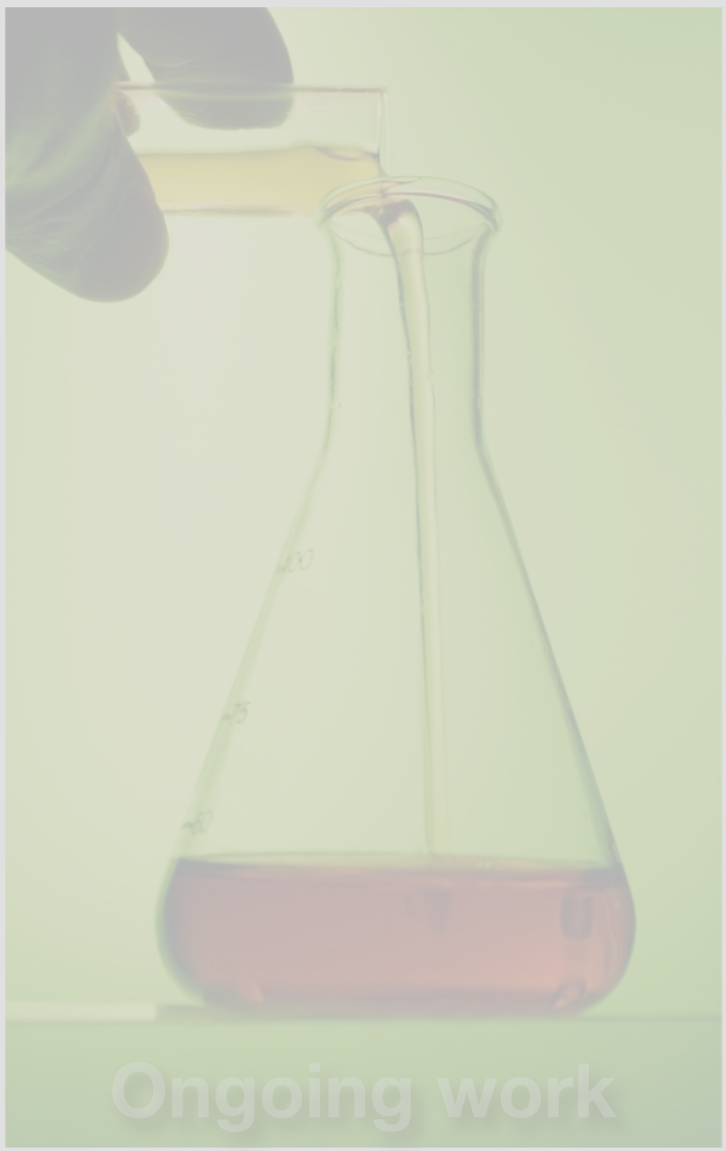
- Tool-supported help for developing and evolving transformations.
- Management, sharing, or documentation of a transformation throughout its entire lifecycle.
- Faster development or reaction to requirement changes.
- Easier transformation development for non experts.

Agenda

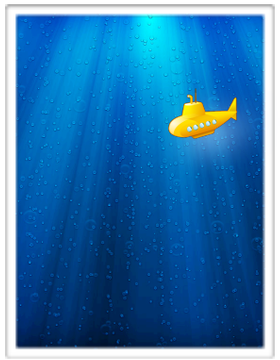


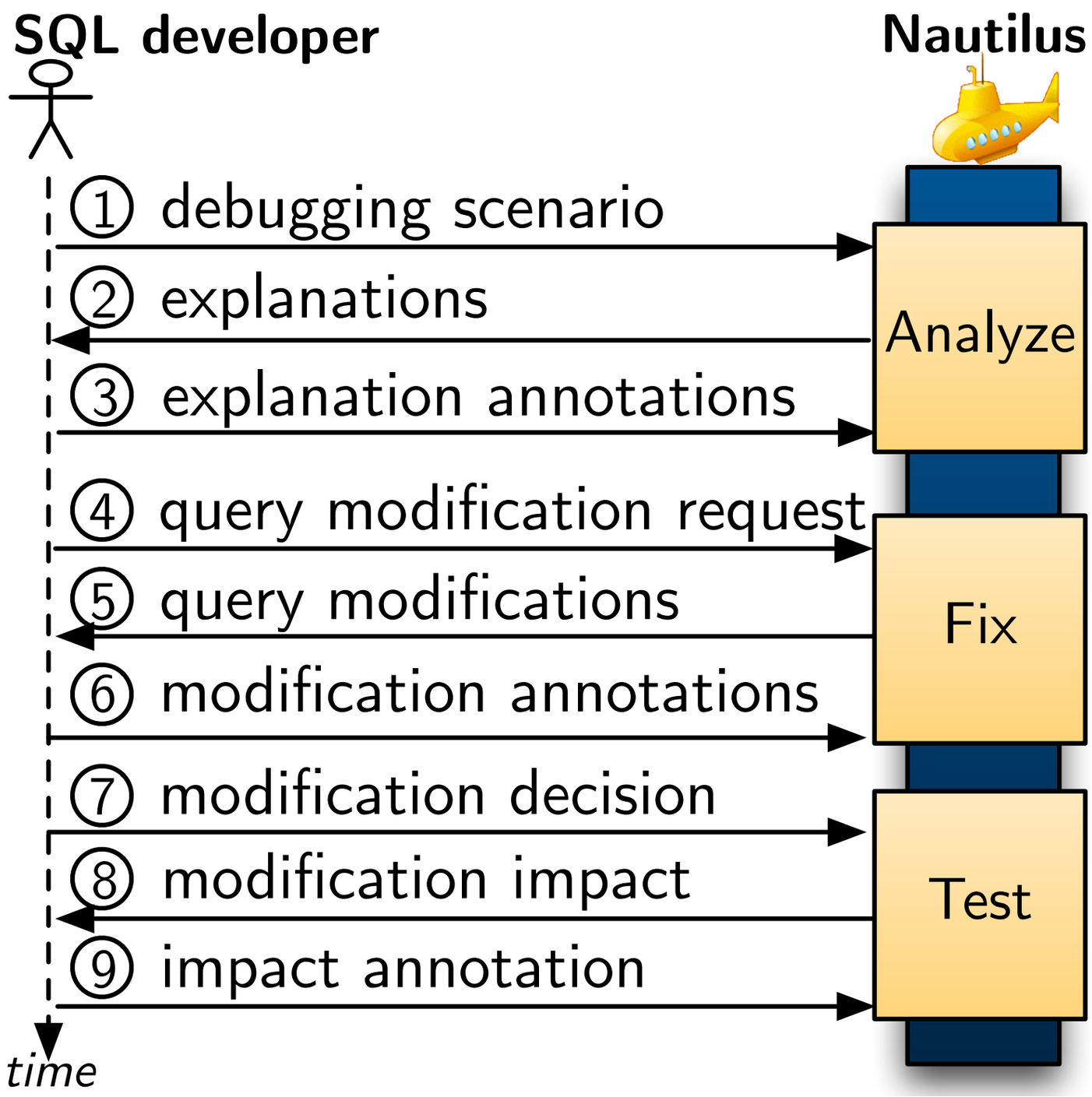
Agenda



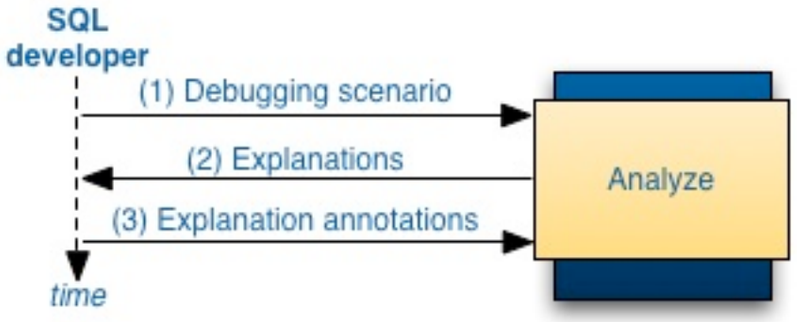


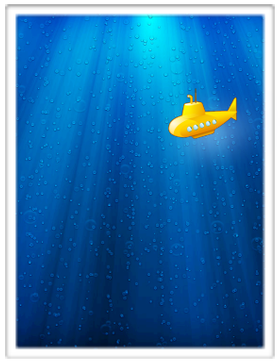
The Complete Workflow





Sample Workflow





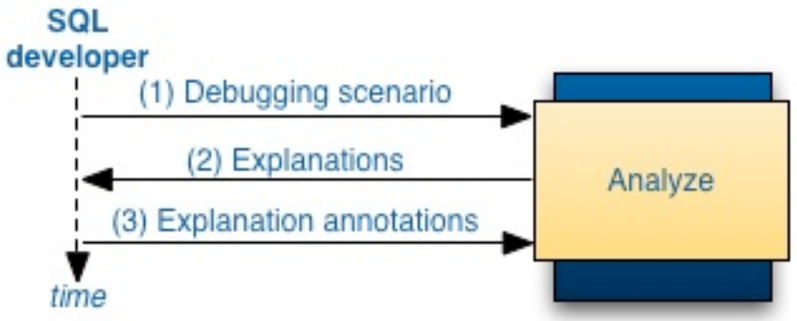
Analyze

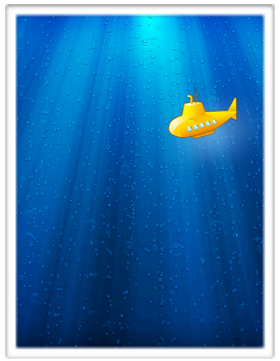
Fix

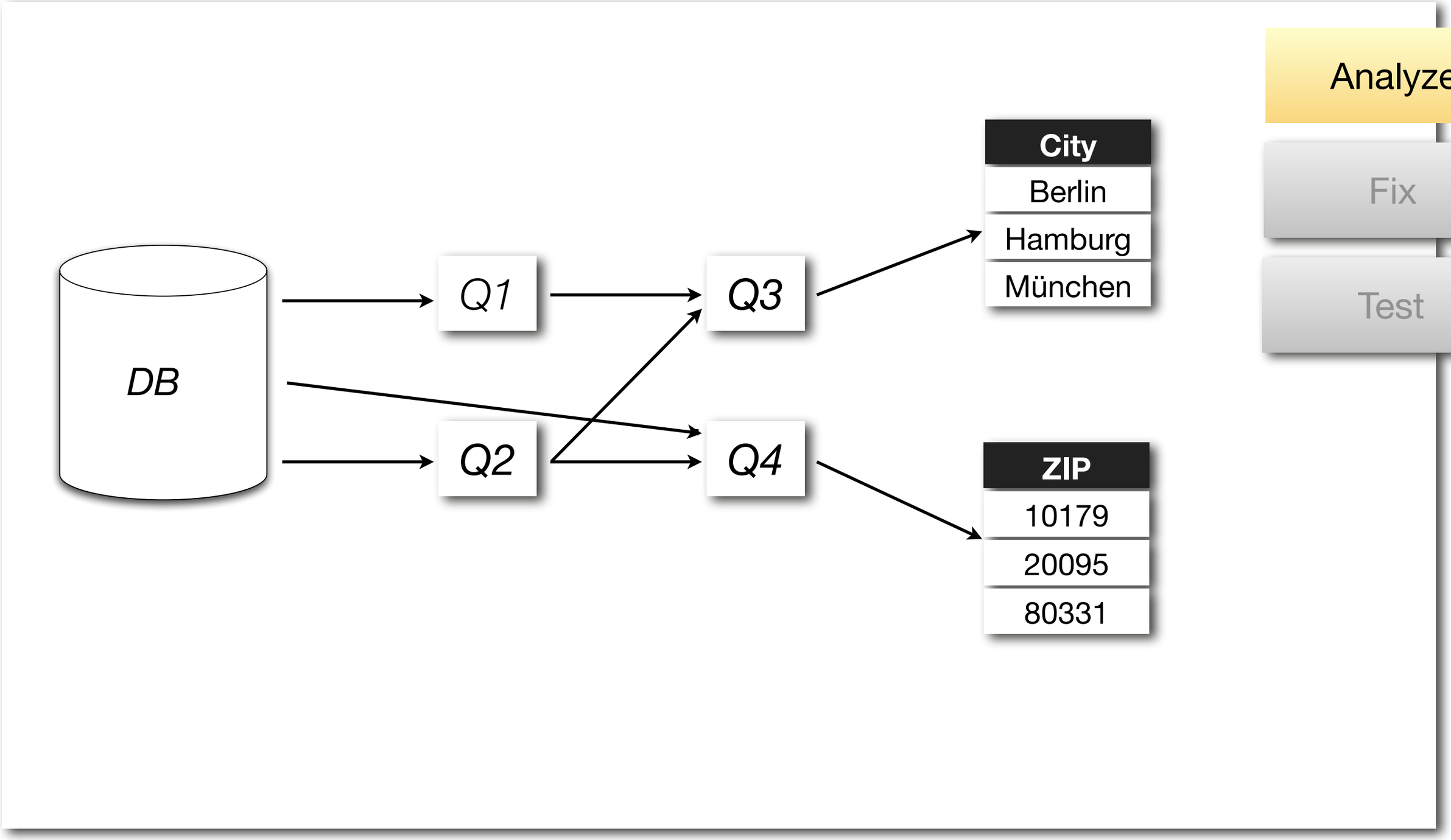
Test

....

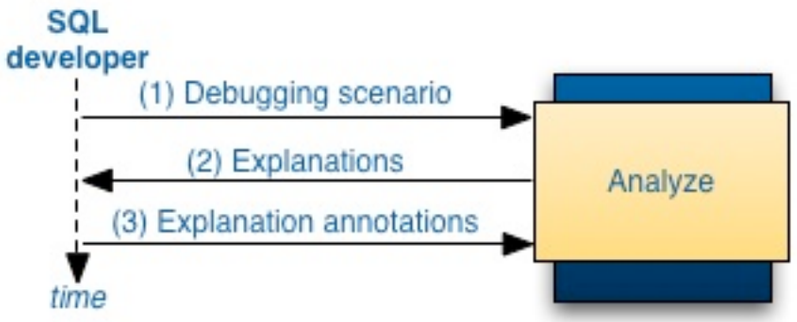
Sample Workflow

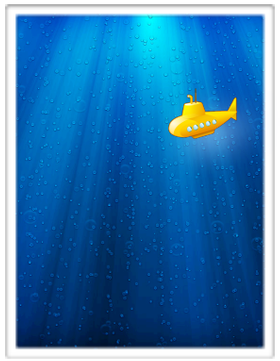


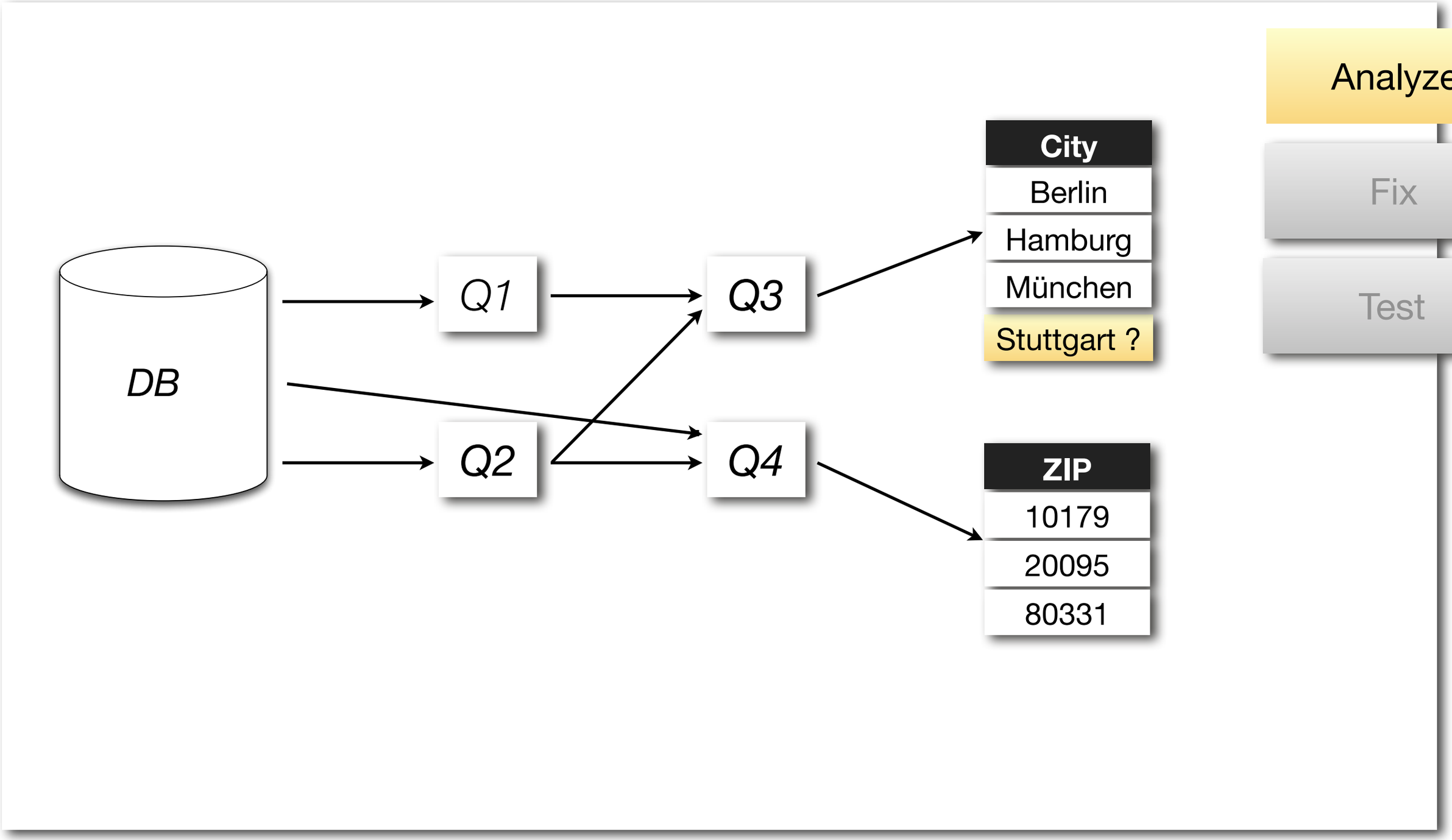




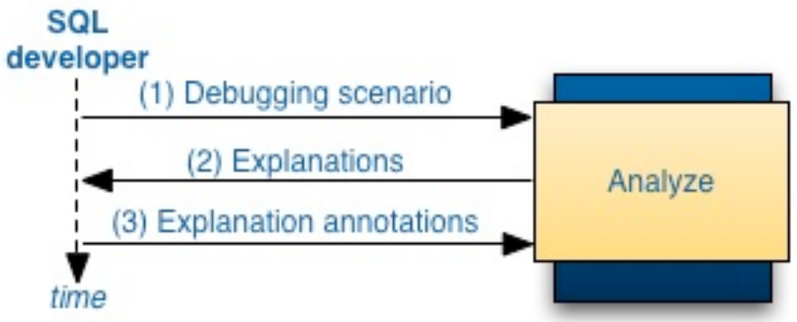
Sample Workflow

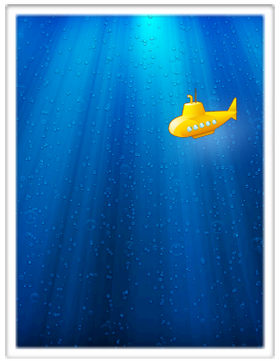


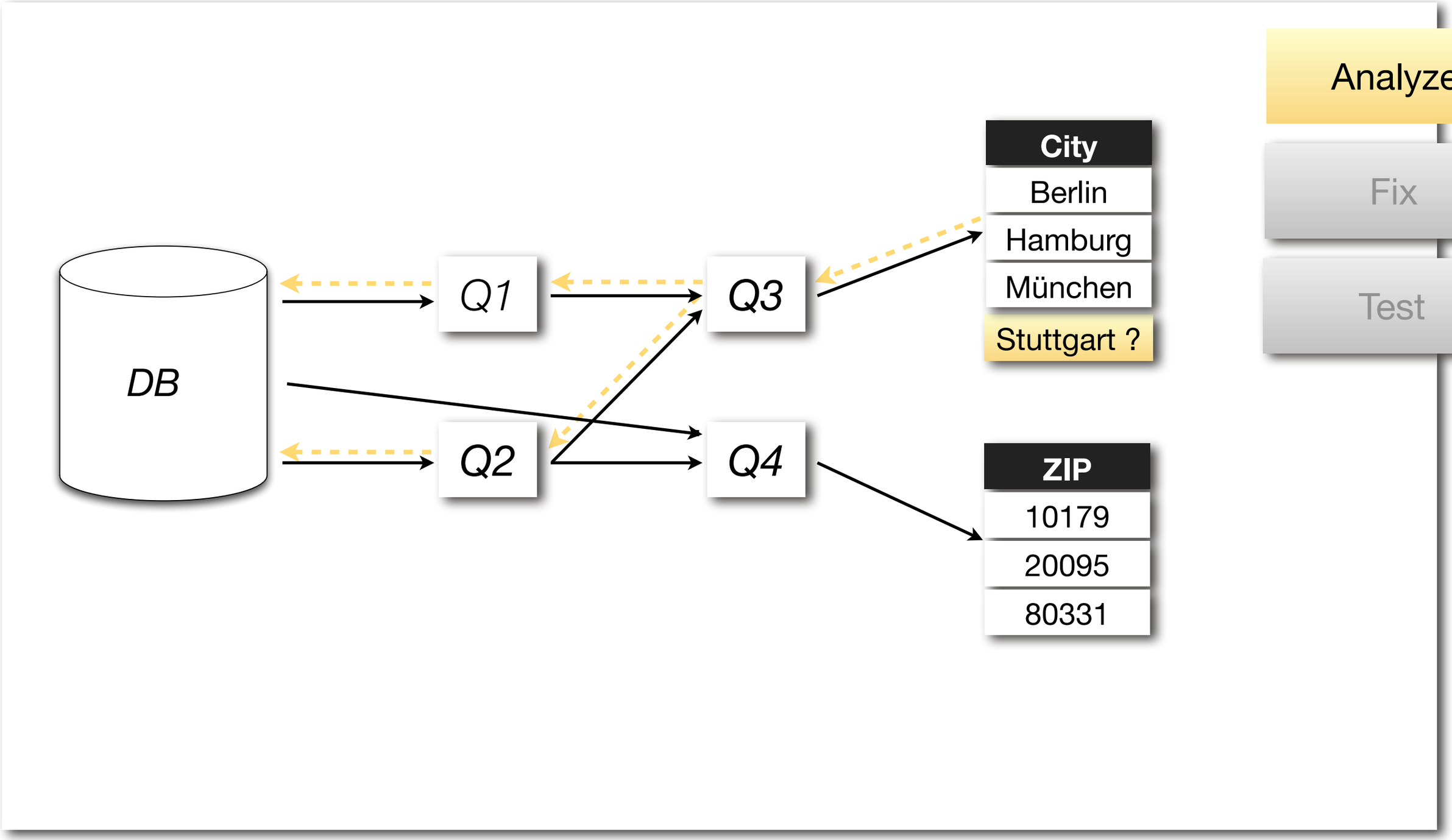




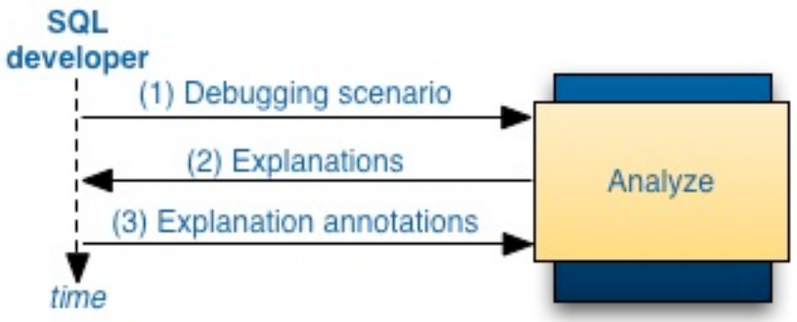
Sample Workflow

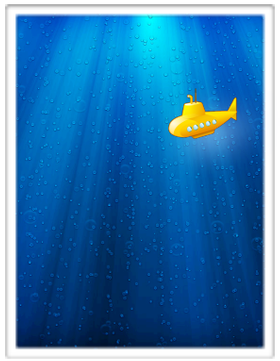






Sample Workflow

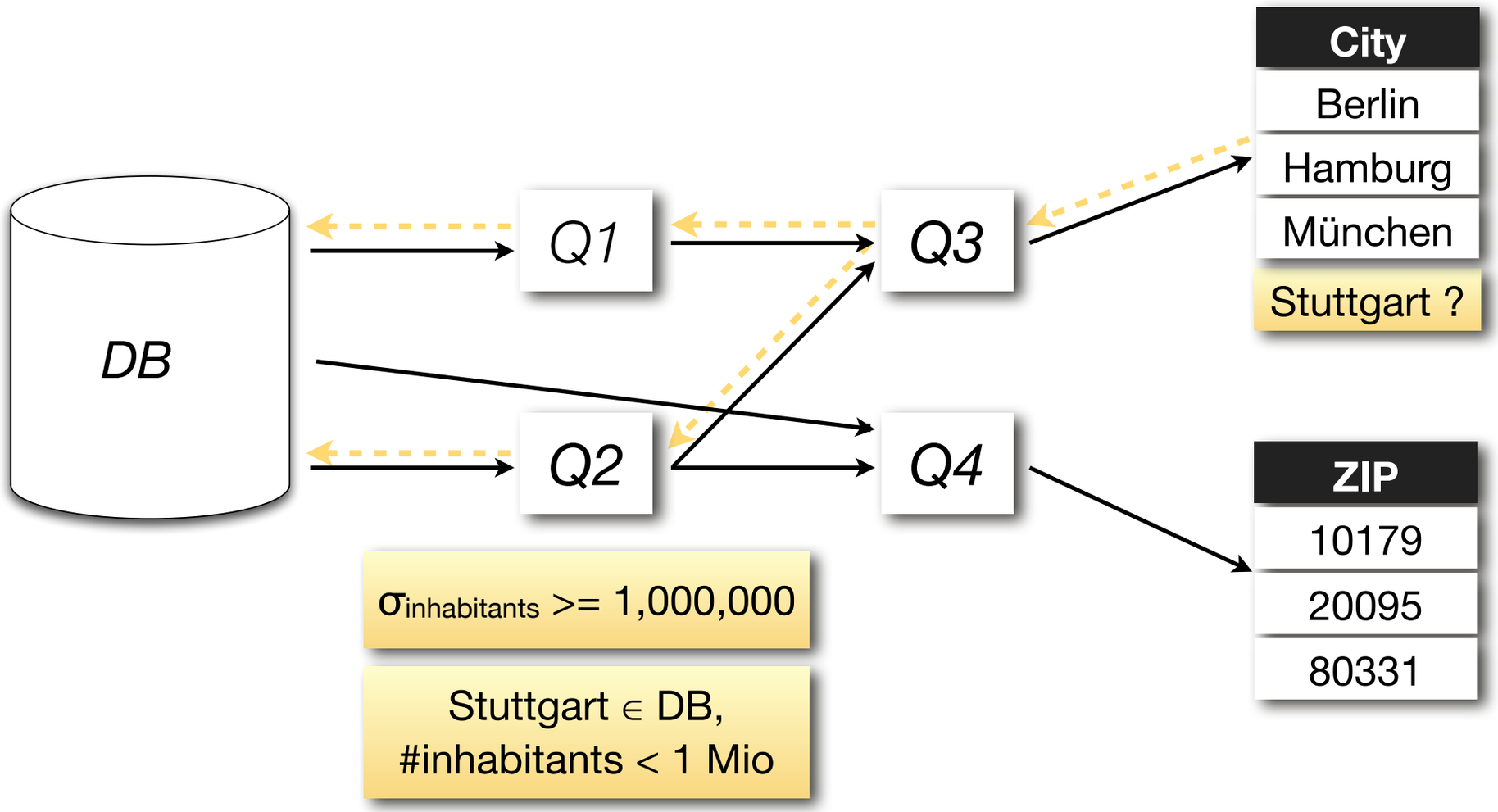




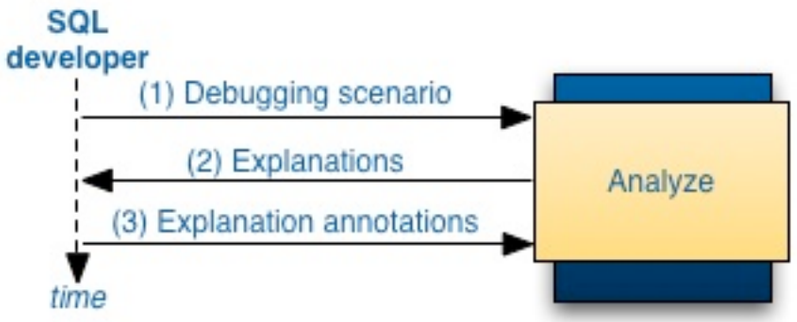
Analyze

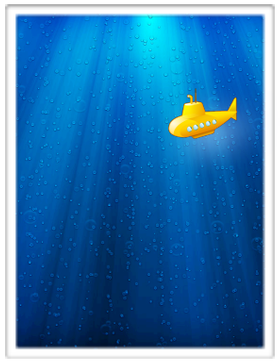
Fix

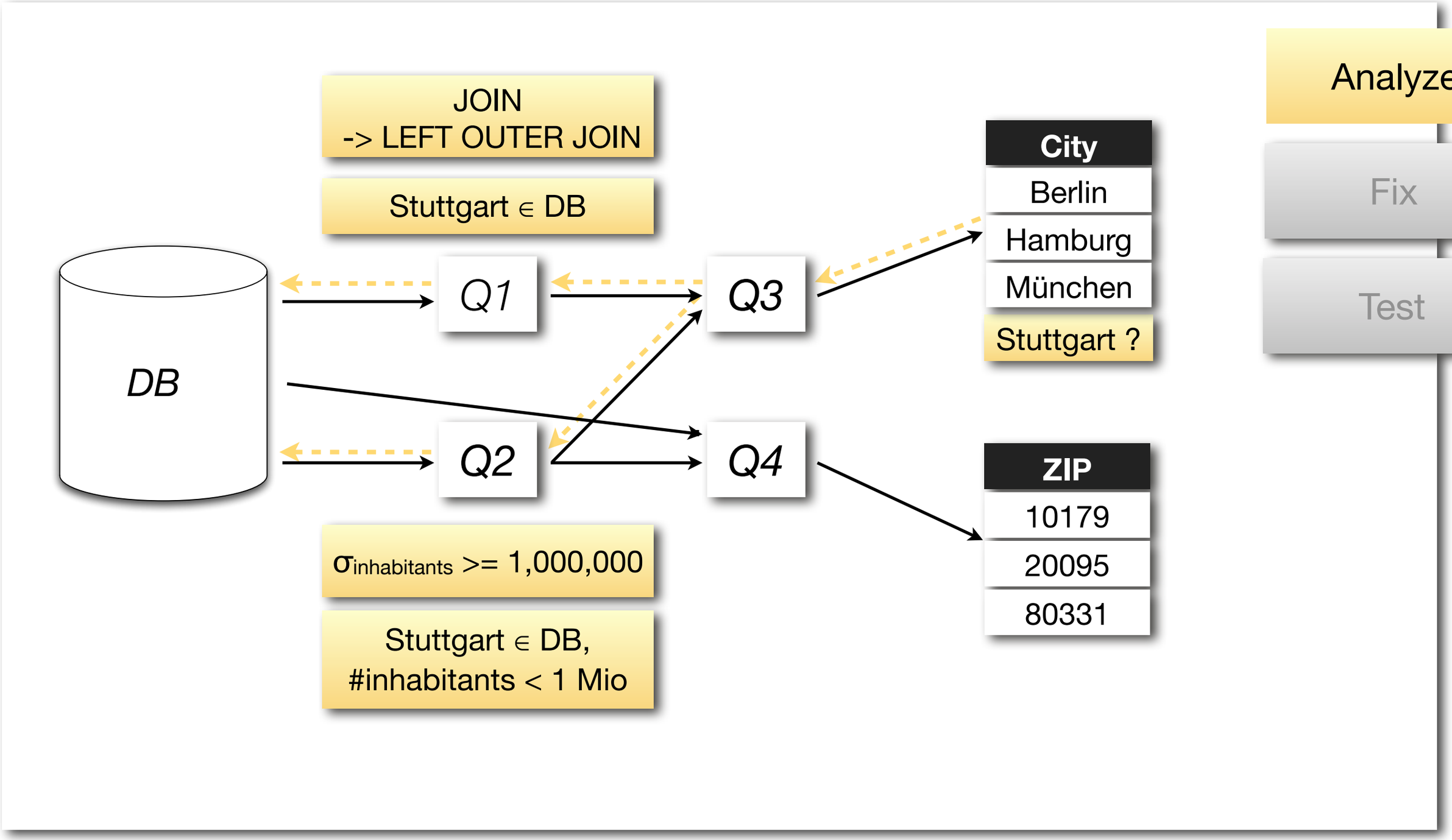
Test



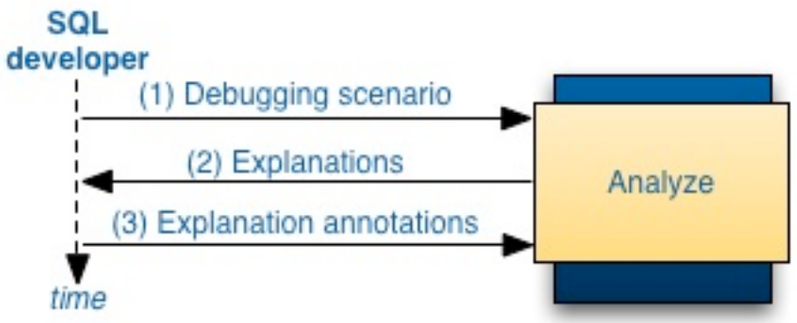
Sample Workflow

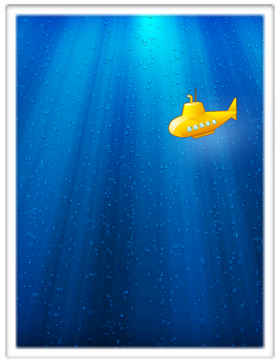






Sample Workflow

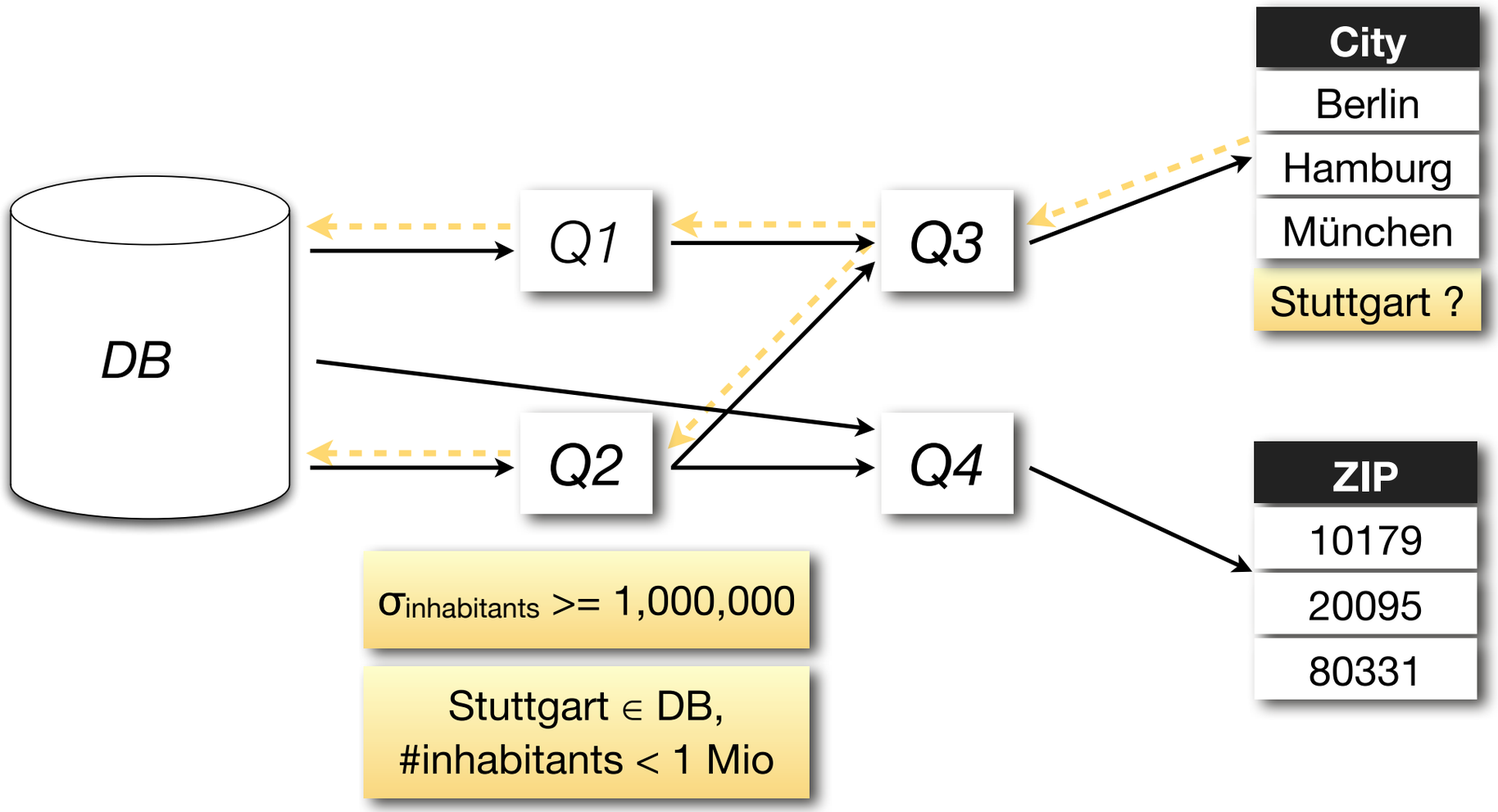




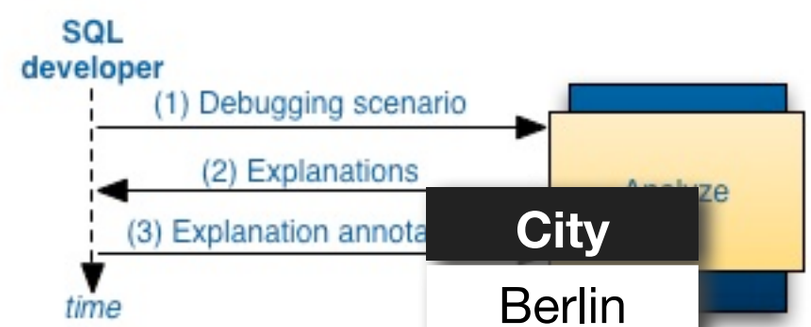
Analyze

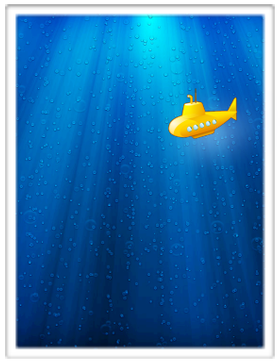
Fix

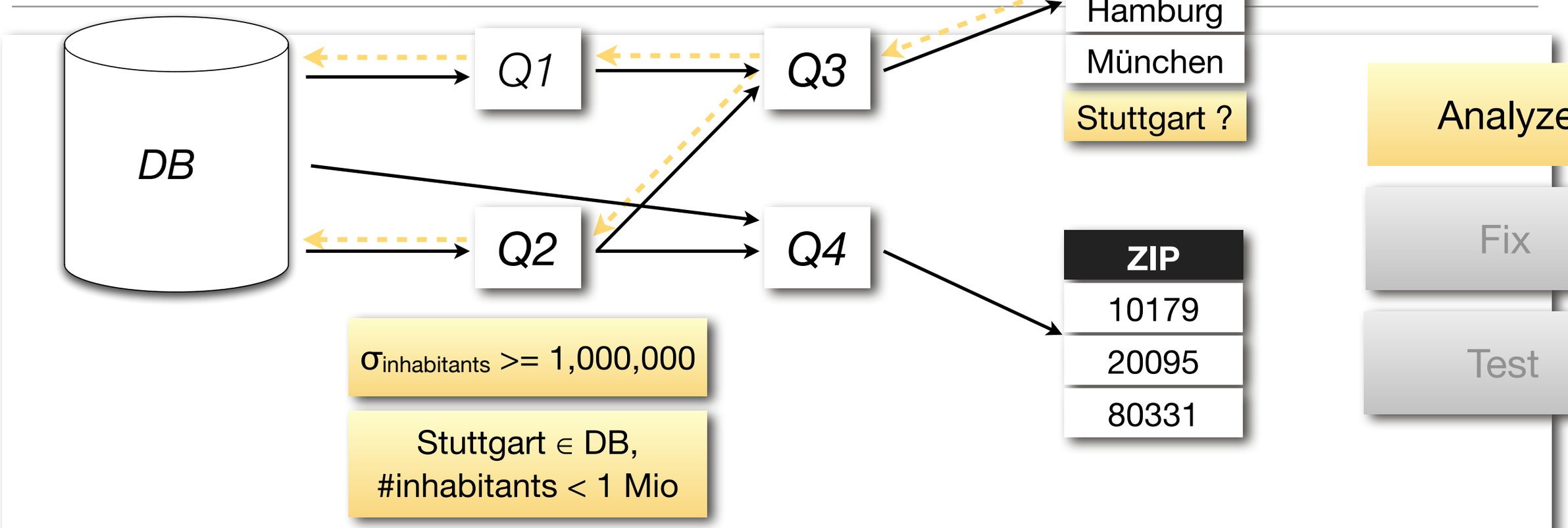
Test



Sample Workflow







(1) Debugging Scenario

- Queries to be analyzed and source data
- Description of the result
- constraints

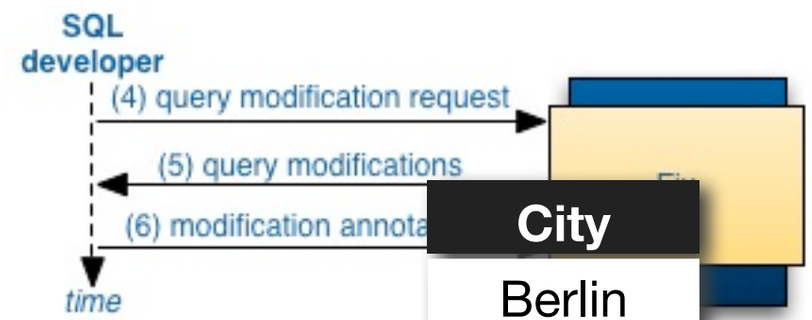
(2) Explanations

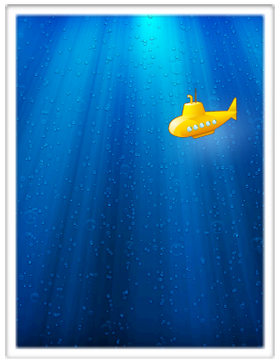
- Explain existing data
- Explain missing data

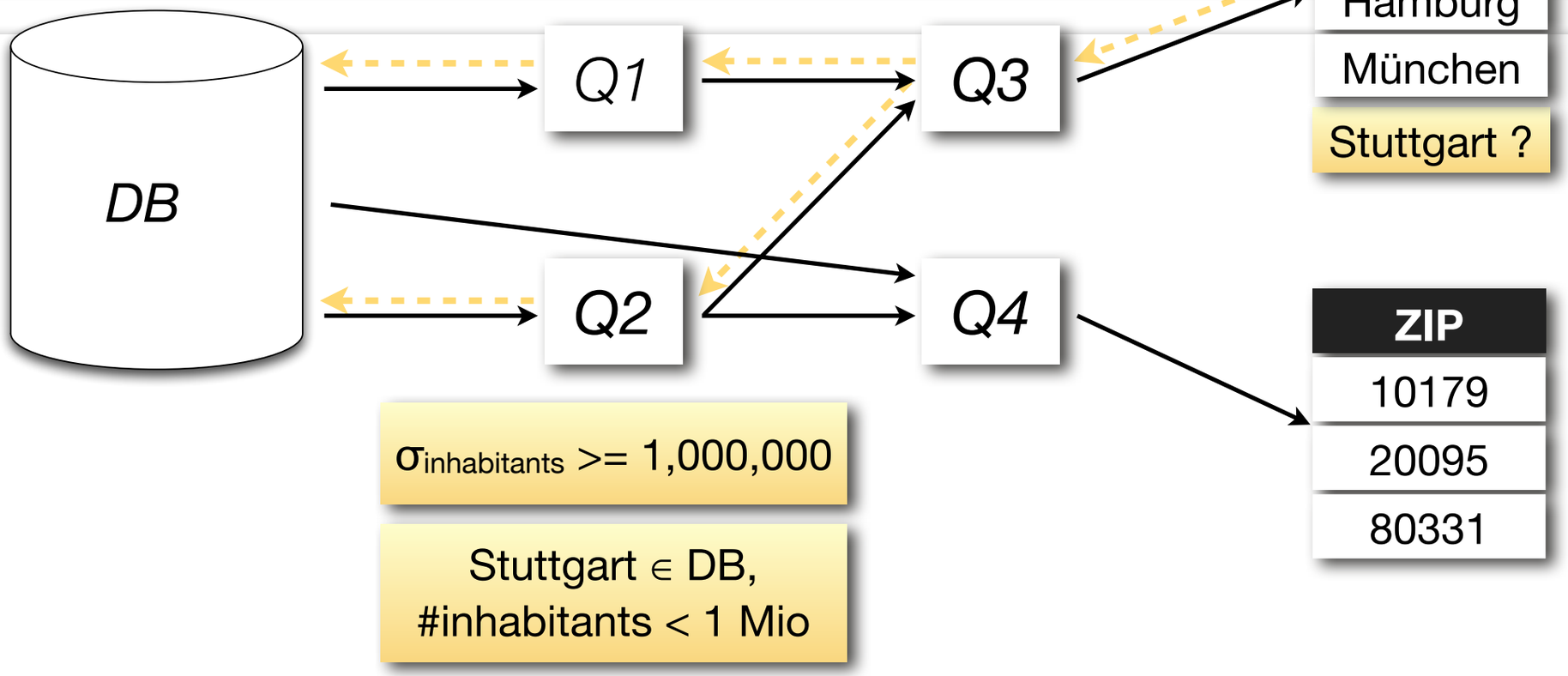
(3) Explanation Annotation

- Guidance and input for fixing phase
- suggestion from the user

Sample Workflow





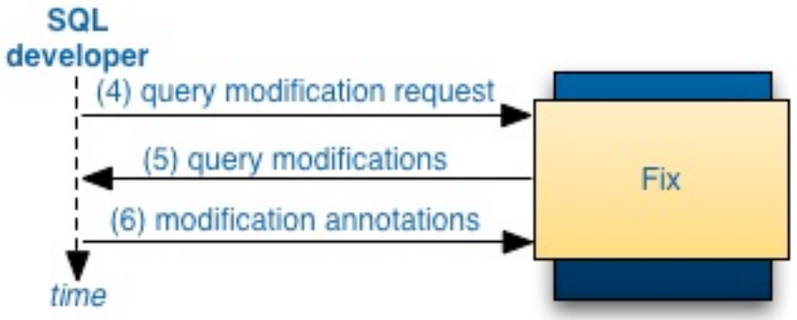


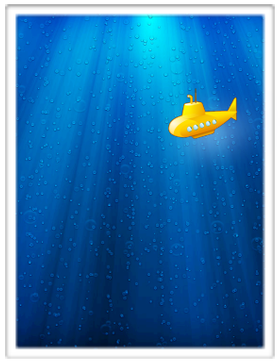
Analyze

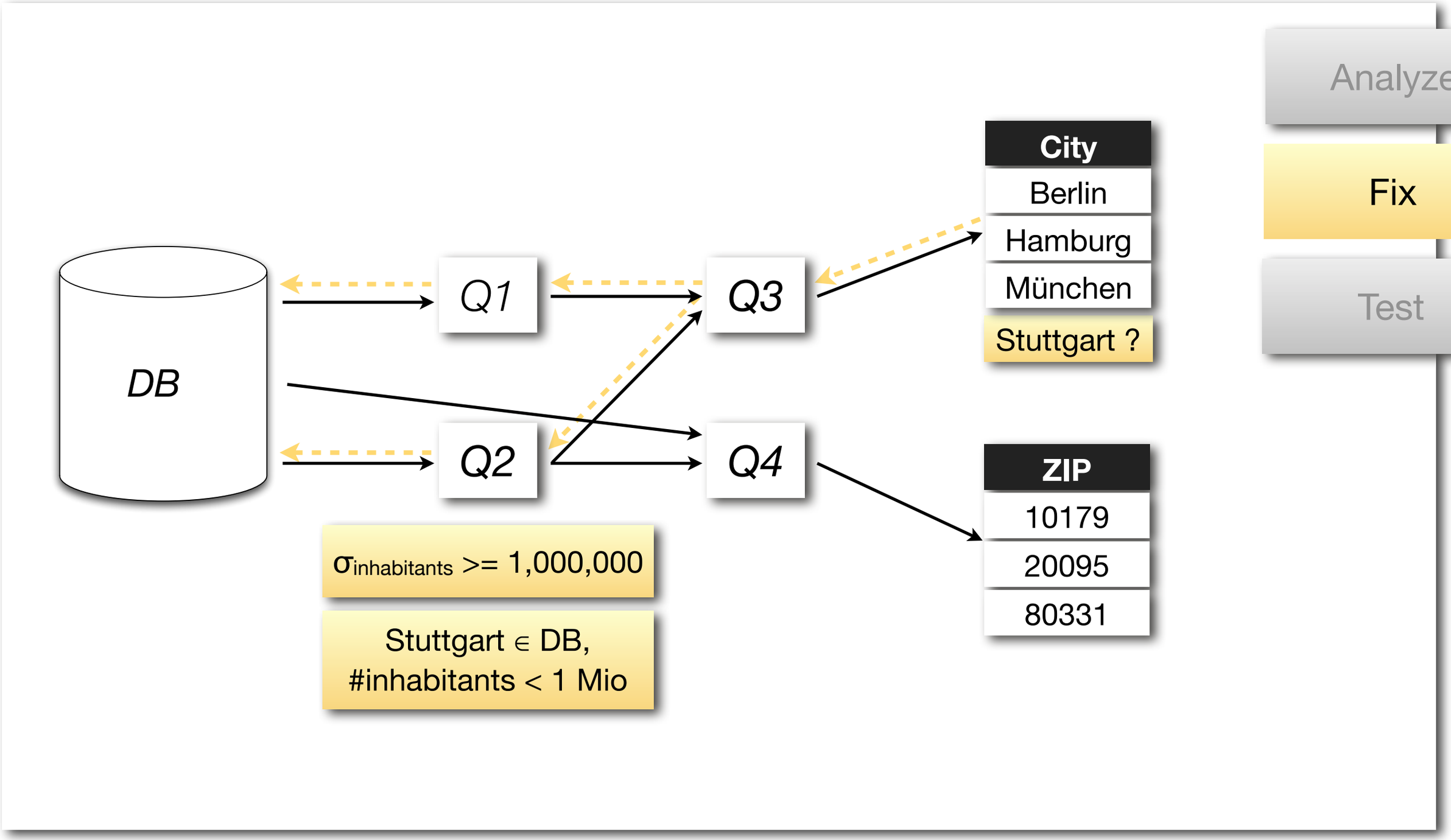
Fix

Test

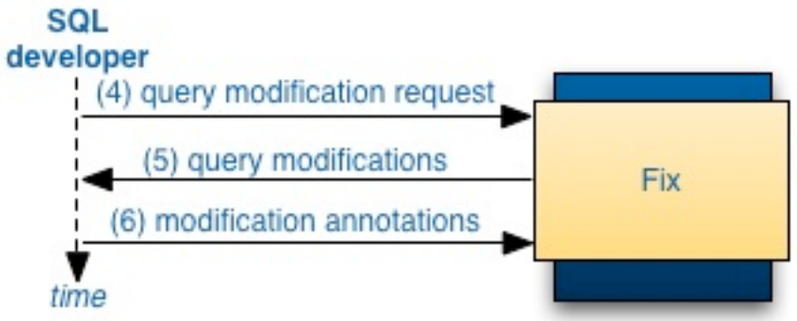
Sample Workflow

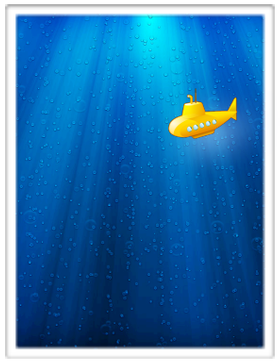






Sample Workflow

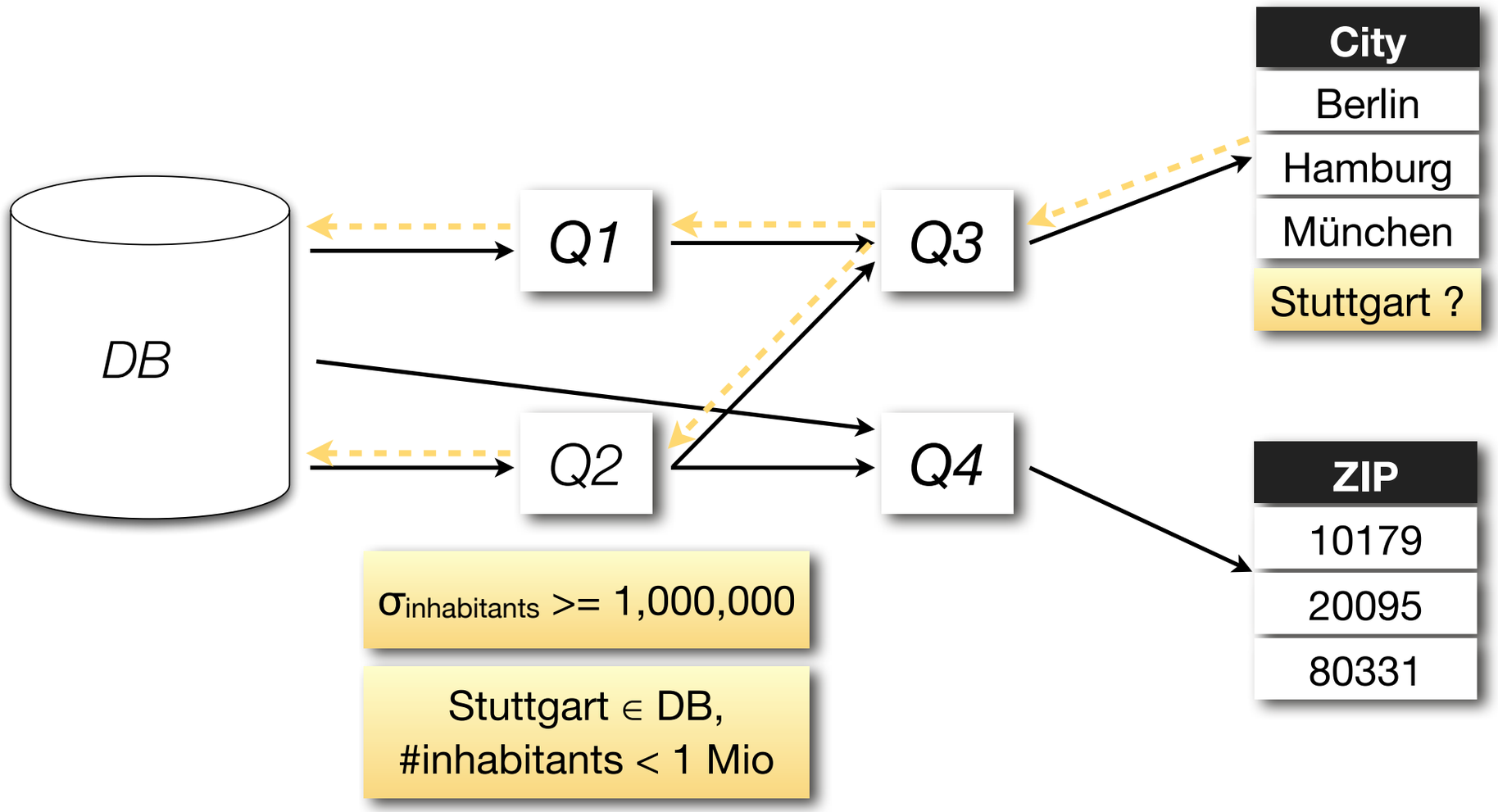




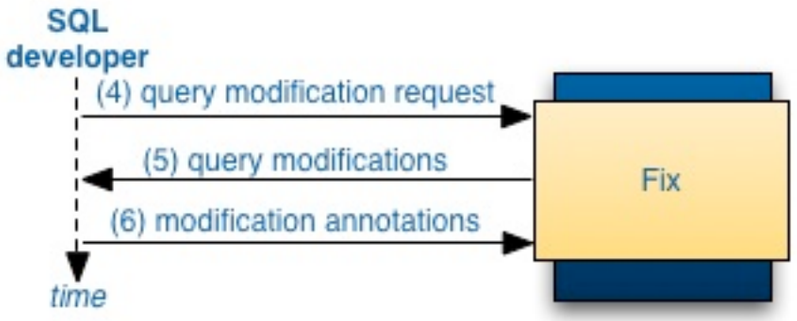
Analyze

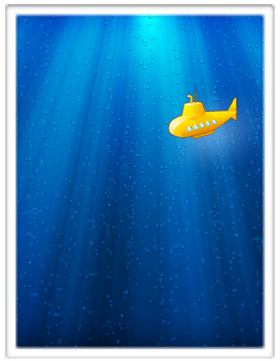
Fix

Test



Sample Workflow

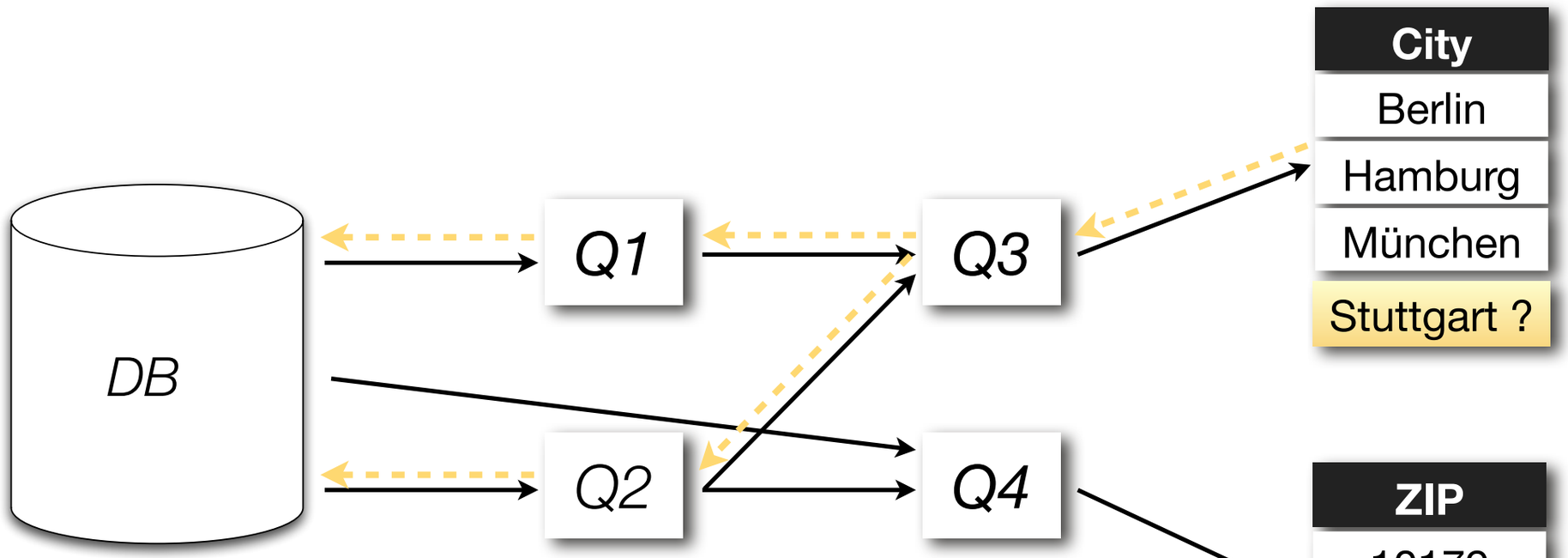




Analyze

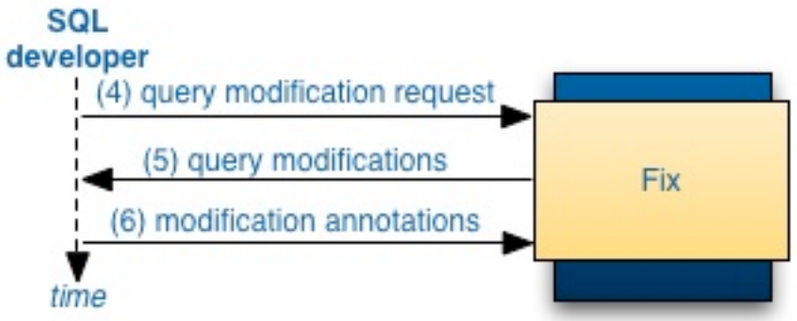
Fix

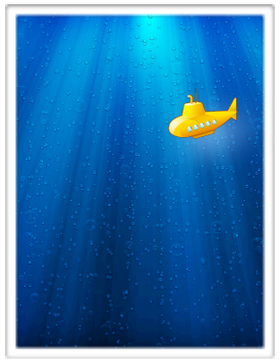
Test

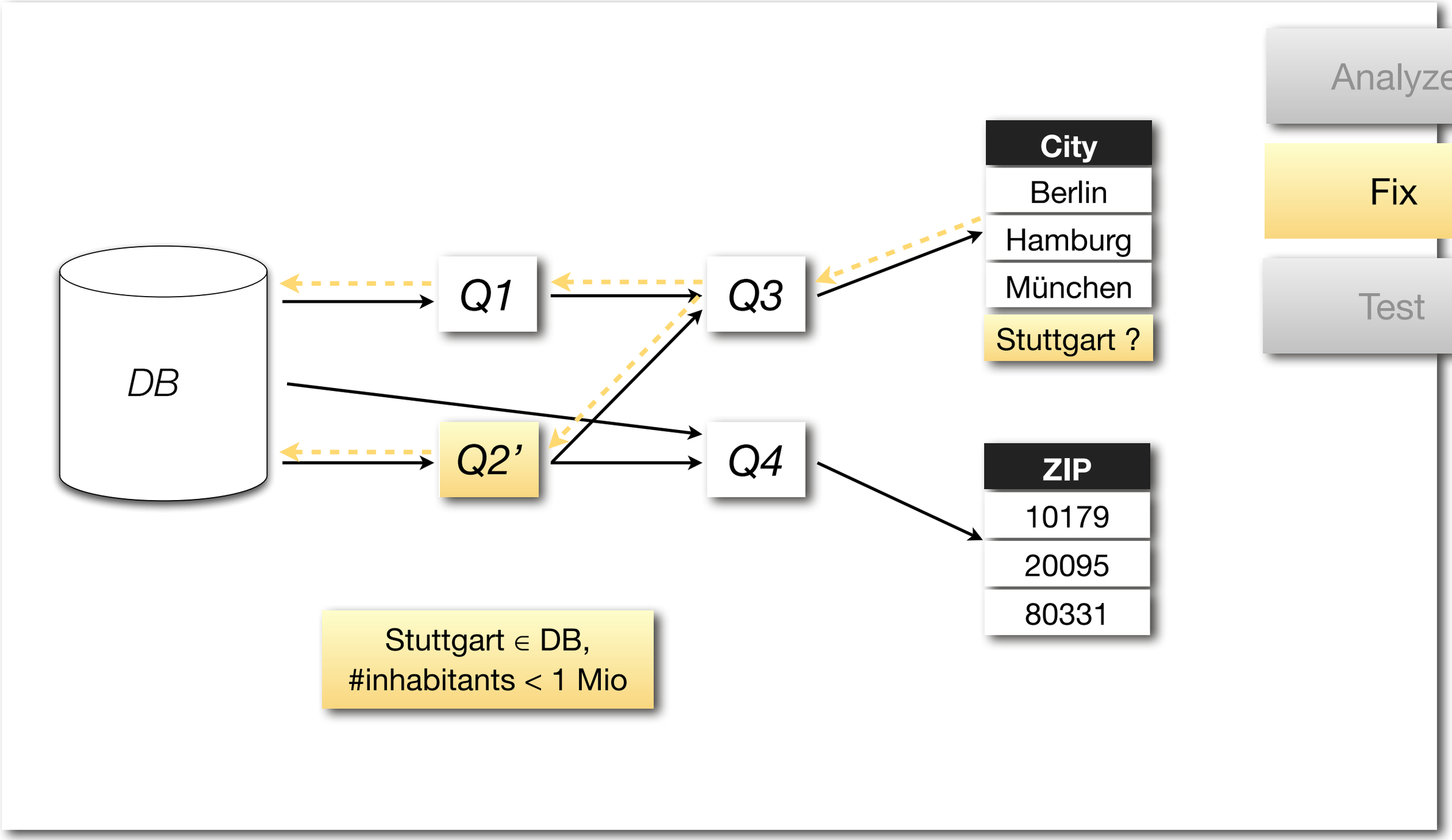


Stuttgart ∈ DB,
#inhabitants < 1 Mio

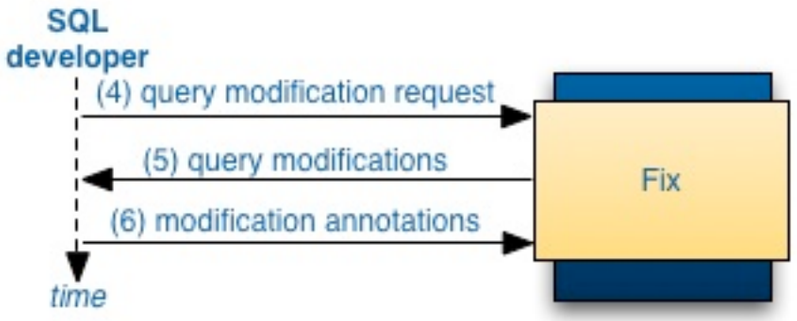
Sample Workflow

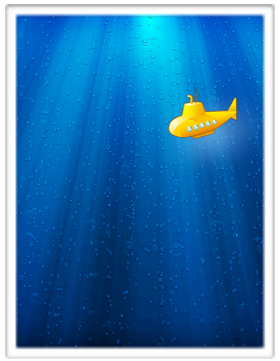






Sample Workflow

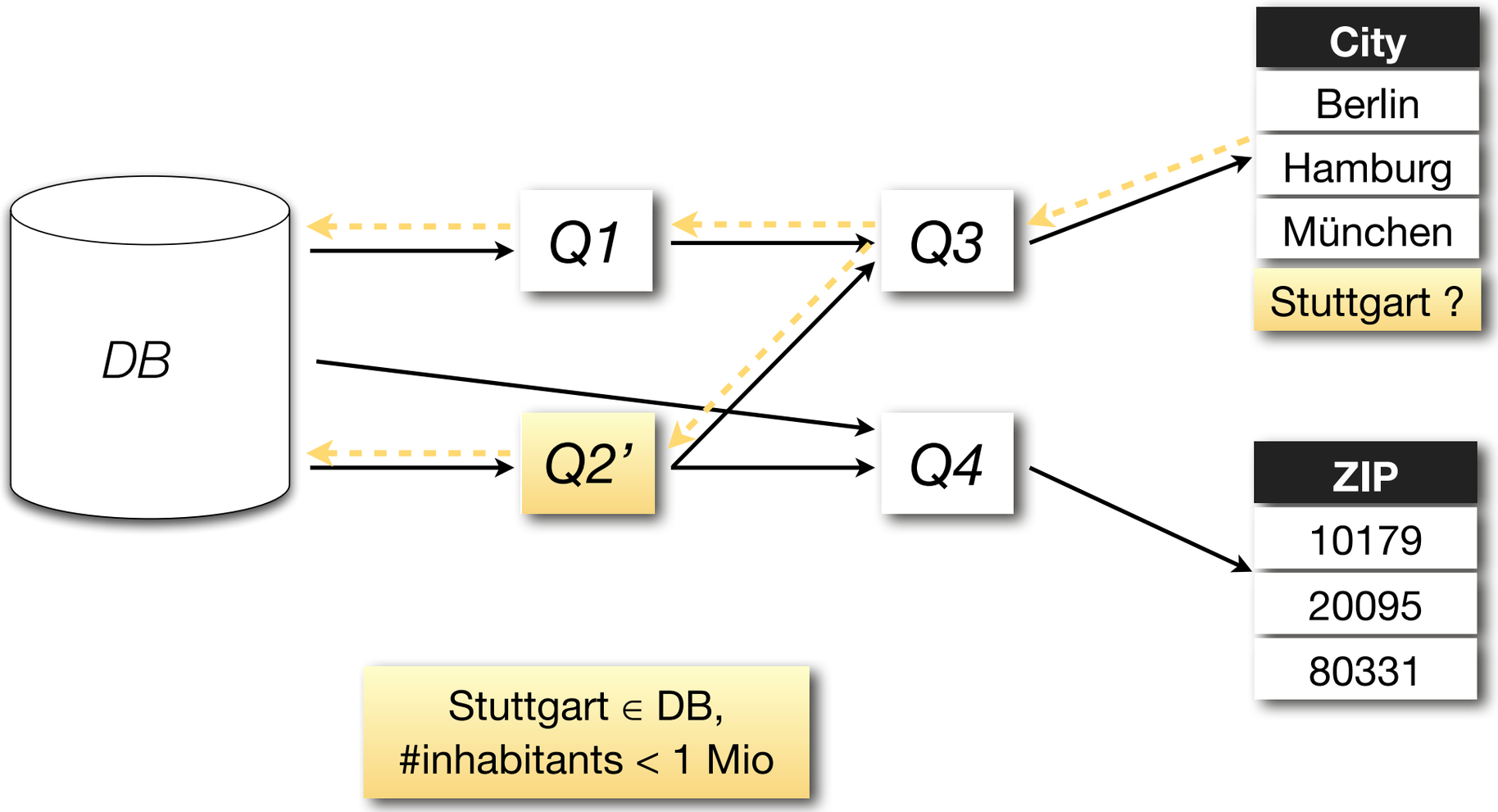




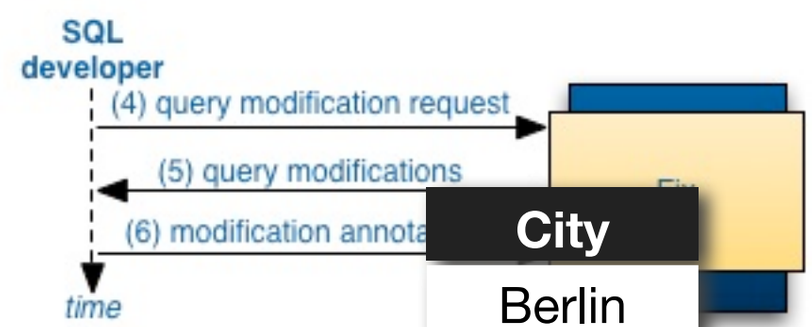
Analyze

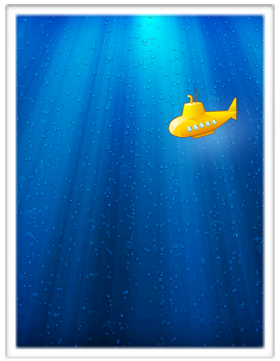
Fix

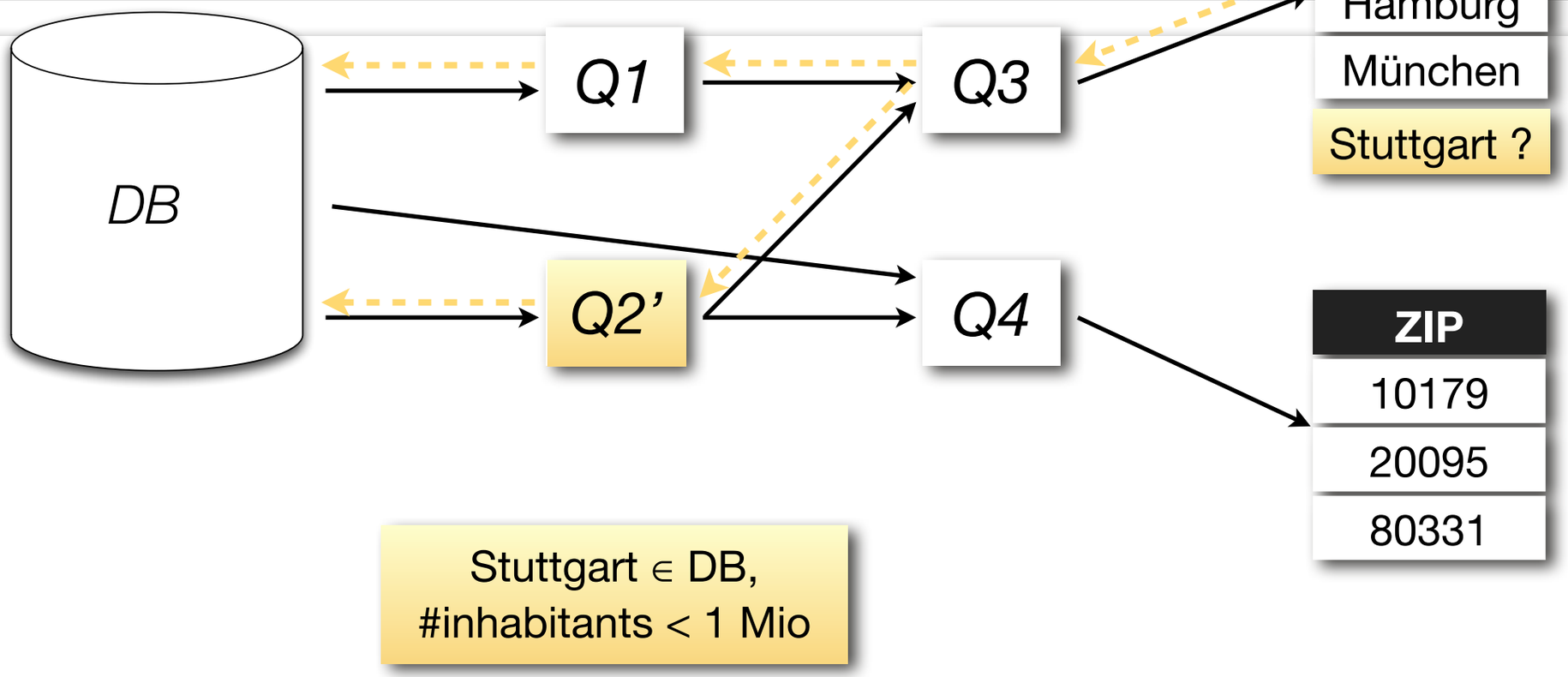
Test



Sample Workflow







Analyze

Fix

Test

query modification request

- request to generate modifications

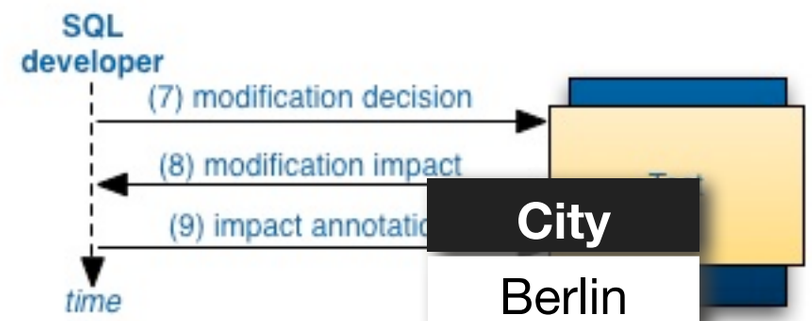
query modifications

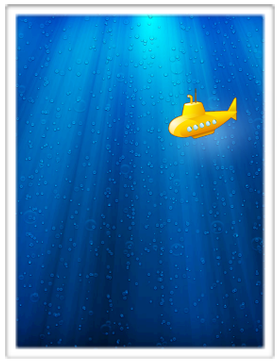
- computing modifications
- w. r. t. previous annotations and constraints
- rewritten SQL with highlighted changes

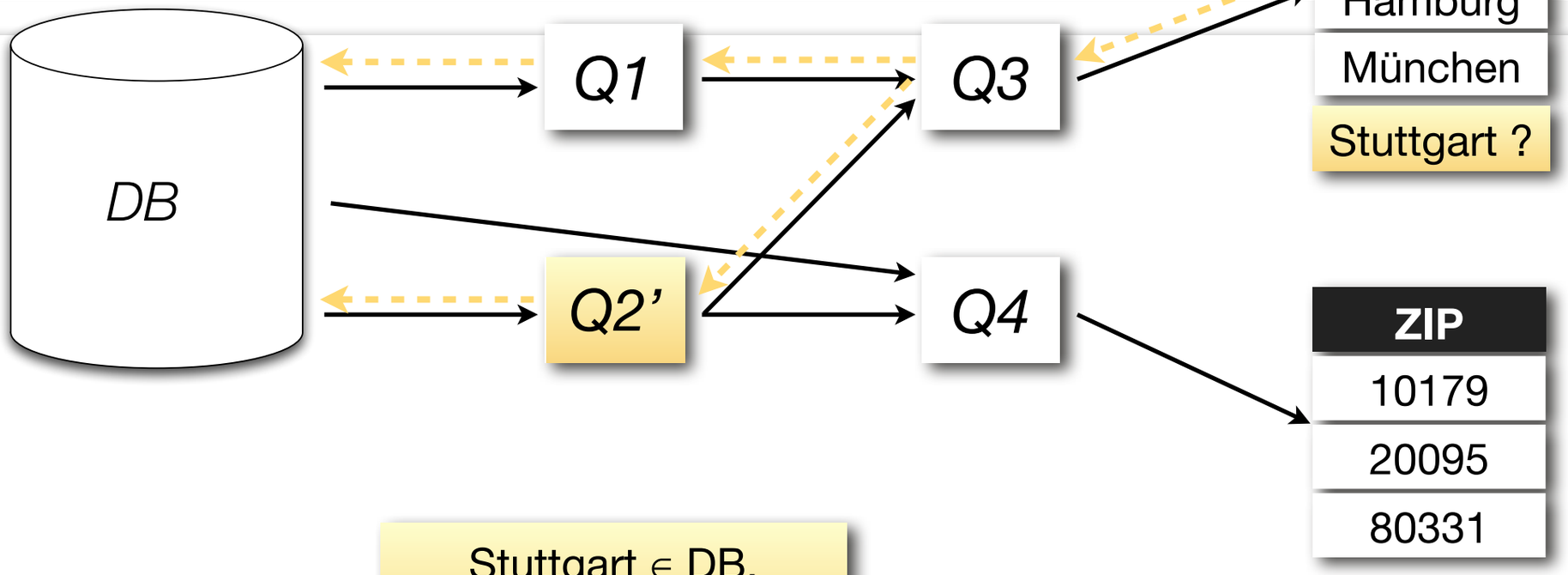
query modification annotation

- analogous to explanation annotation

Sample Workflow







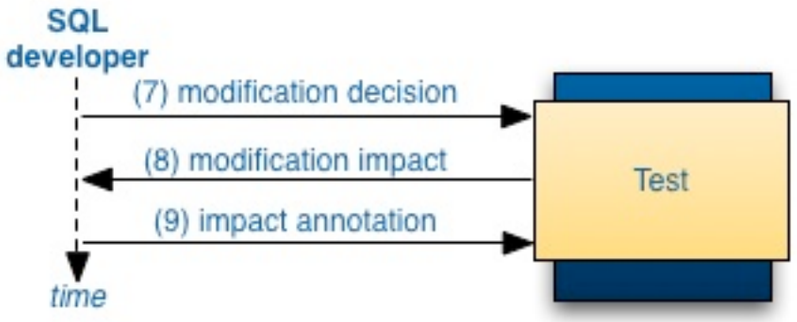
Stuttgart ∈ DB,
#inhabitants < 1 Mio

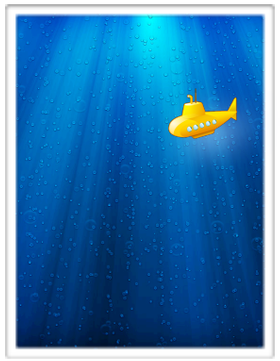
Analyze

Fix

Test

Sample Workflow

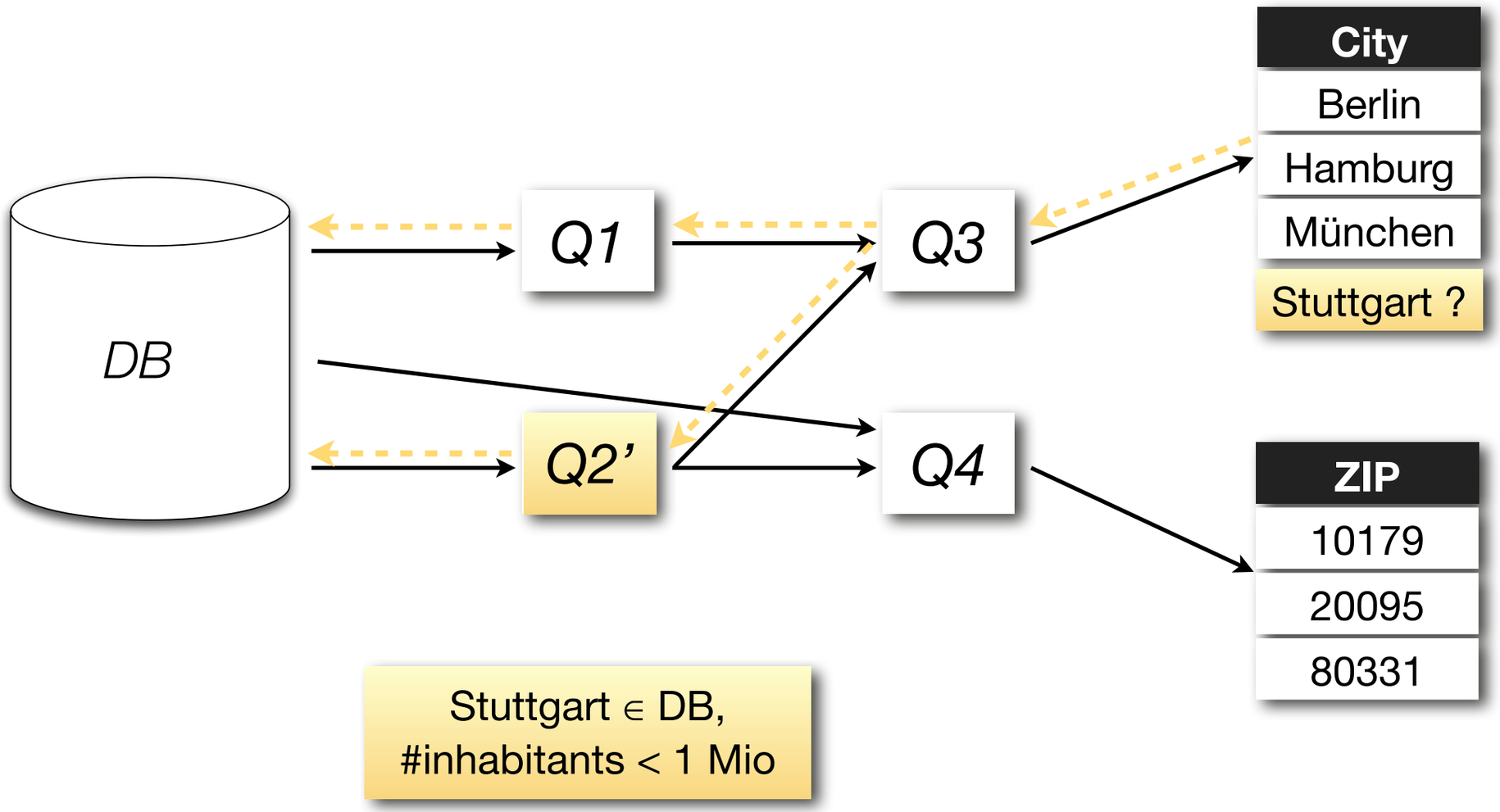




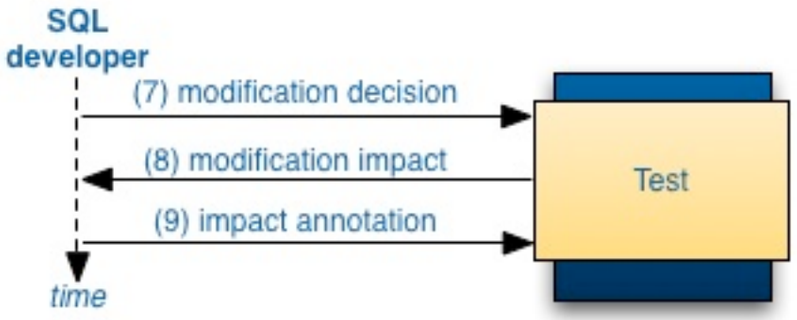
Analyze

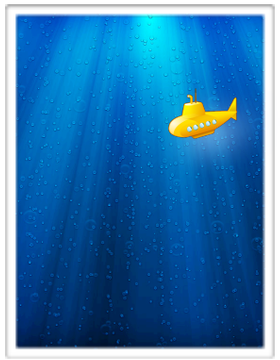
Fix

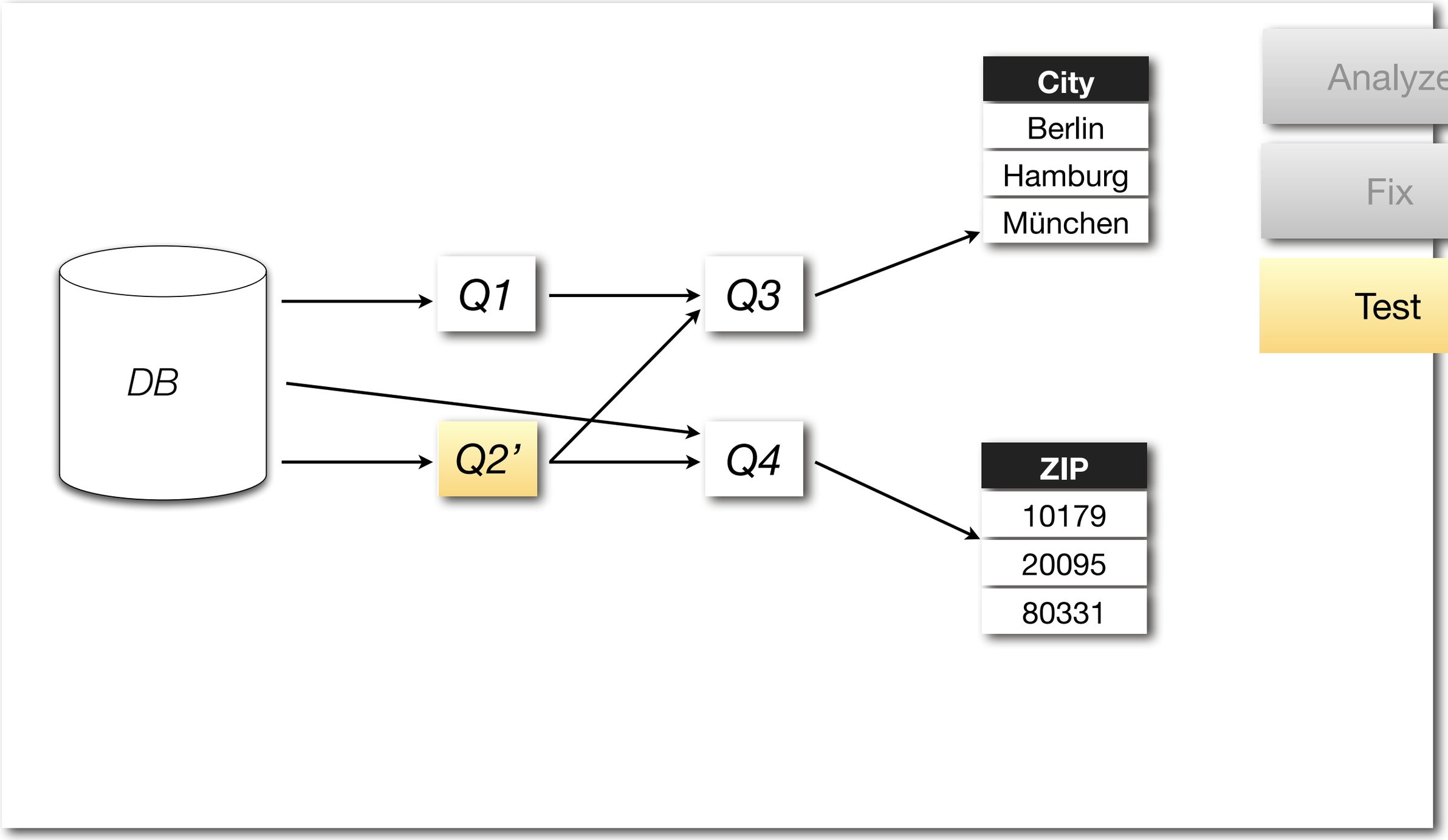
Test



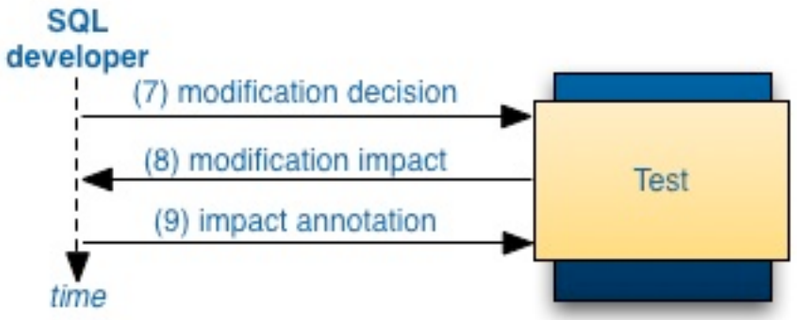
Sample Workflow

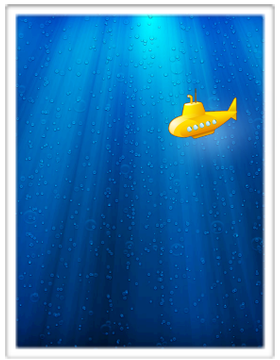


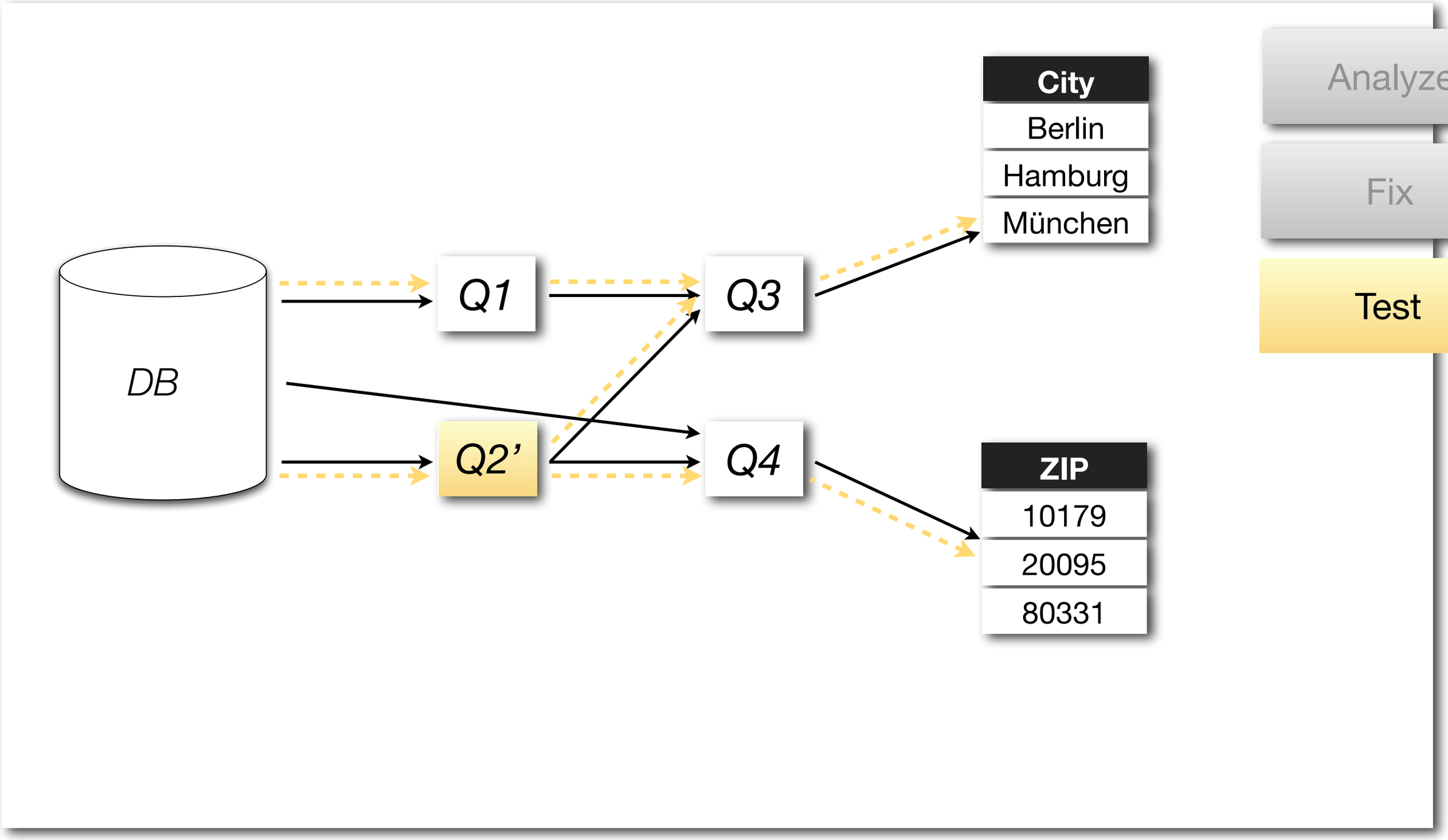




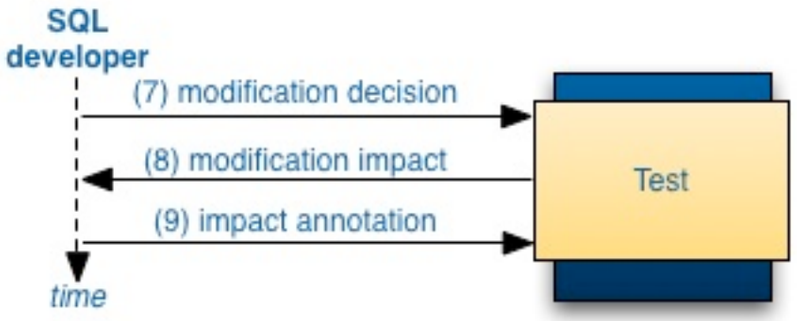
Sample Workflow

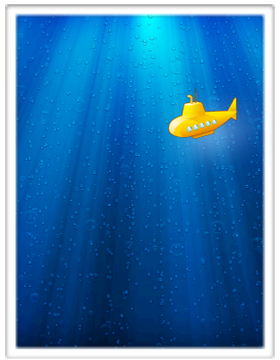


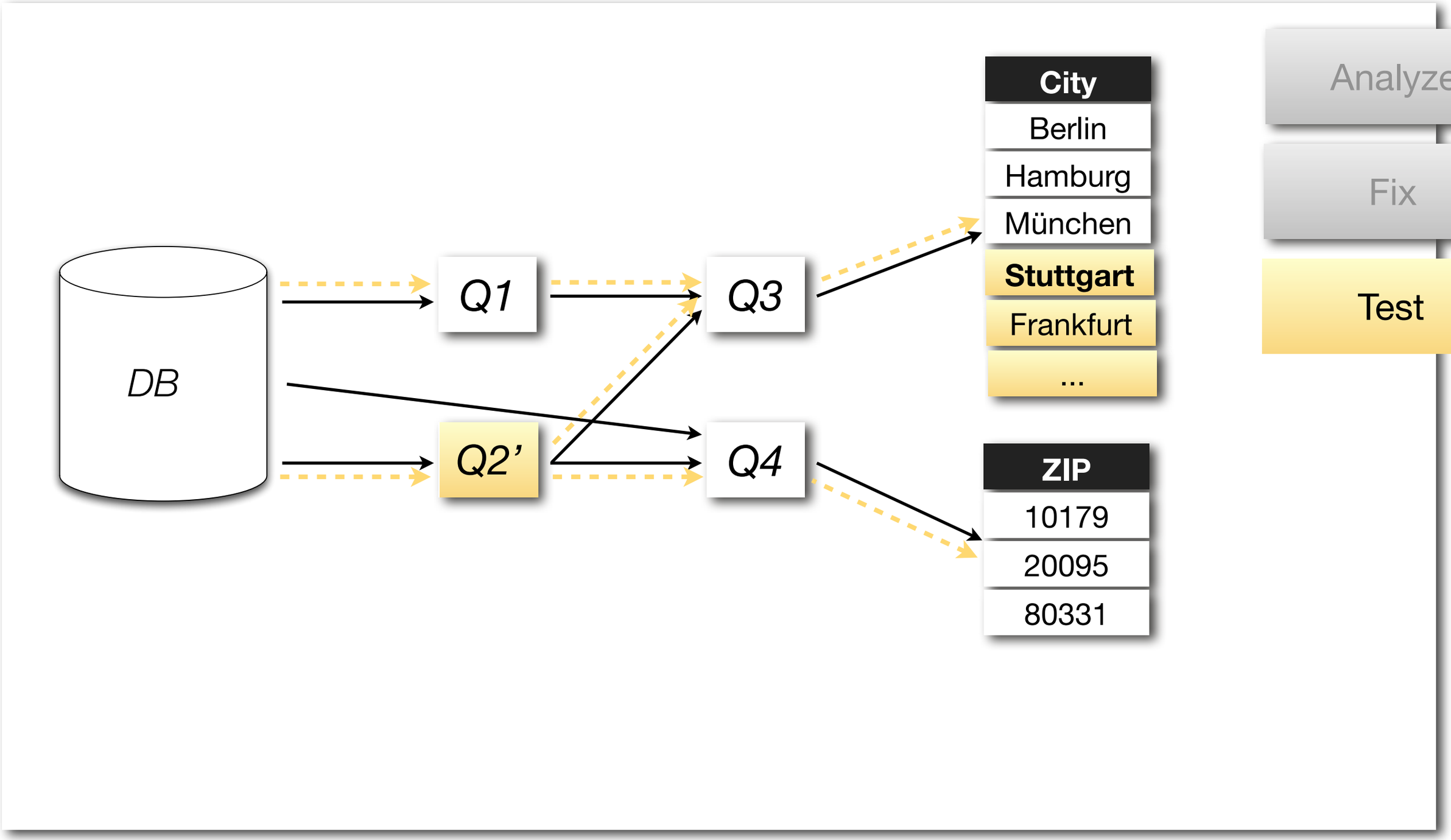




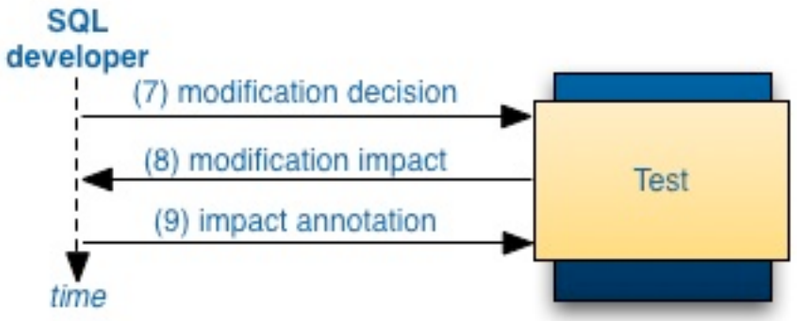
Sample Workflow

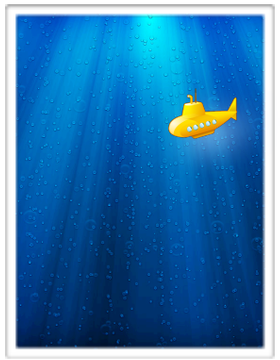


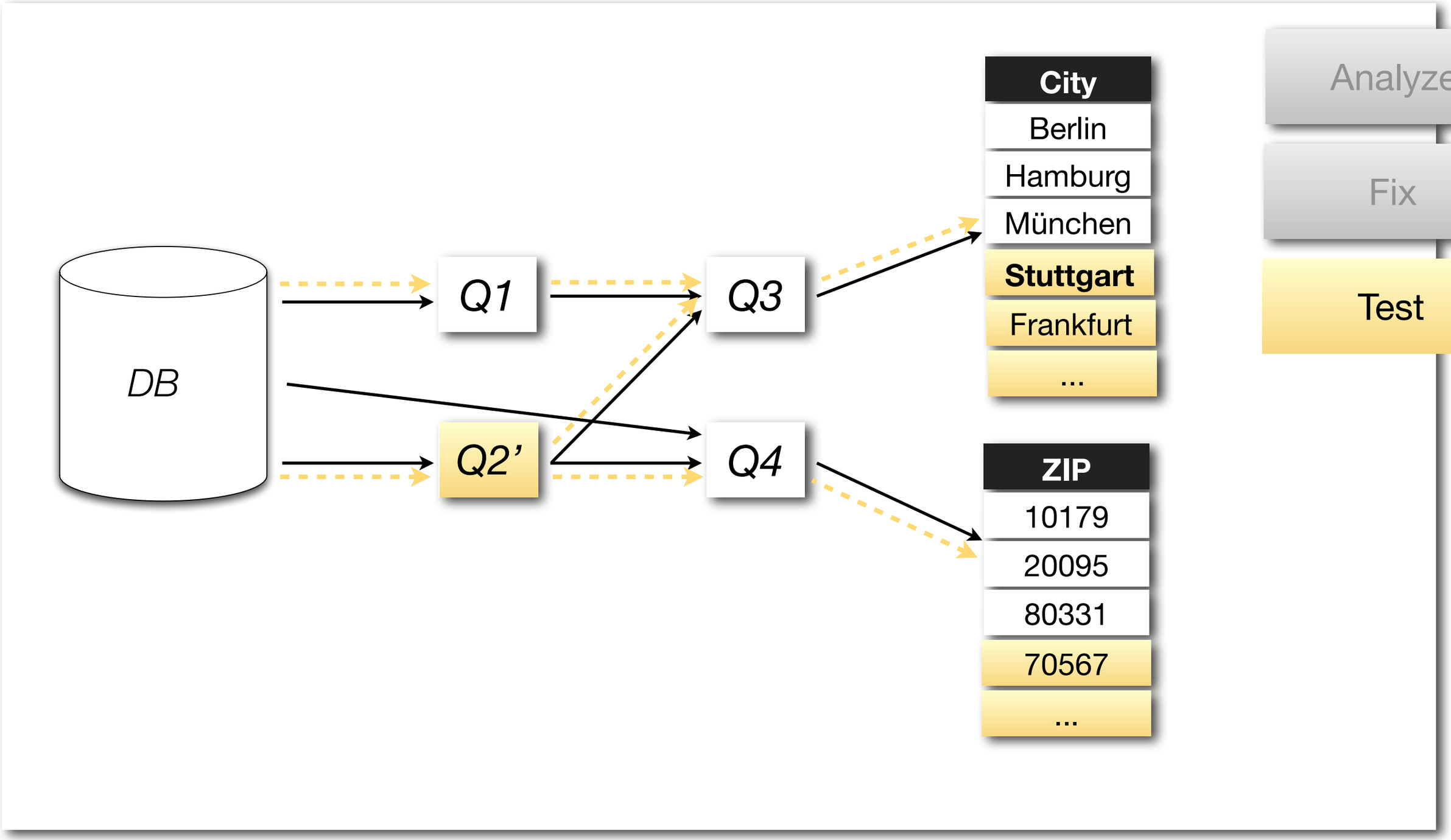




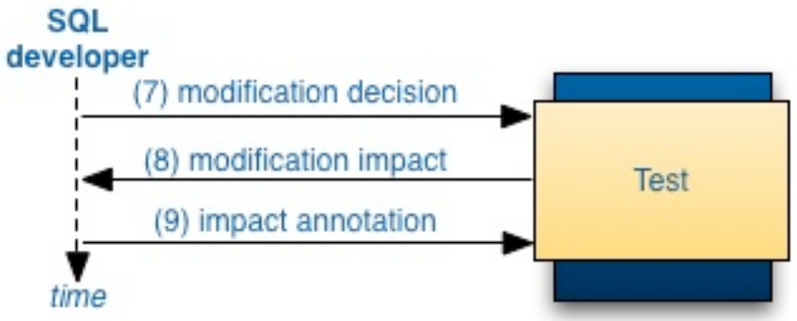
Sample Workflow

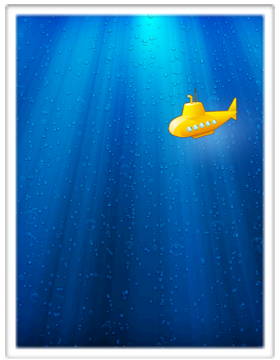


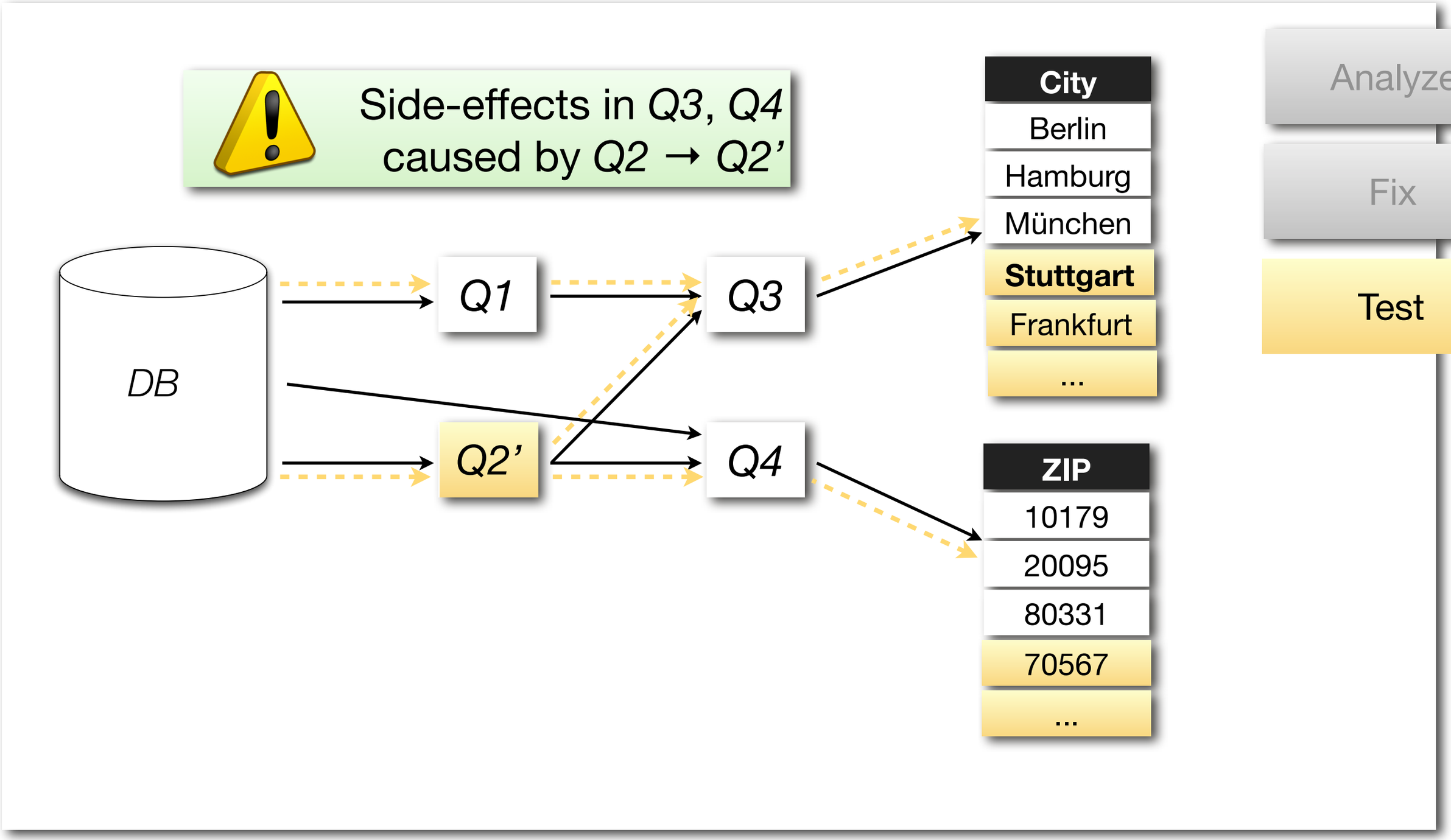




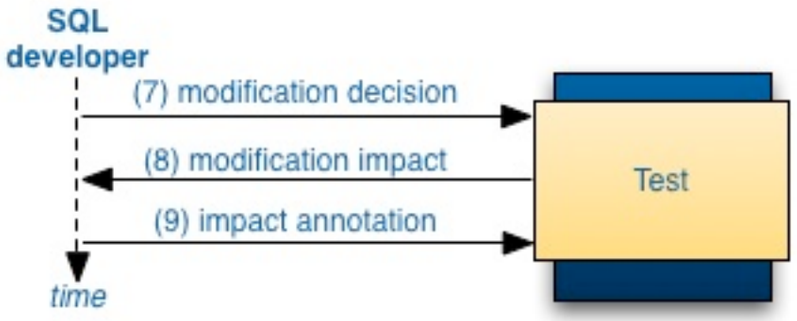
Sample Workflow

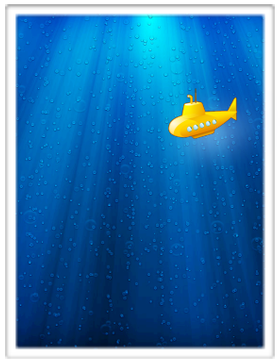




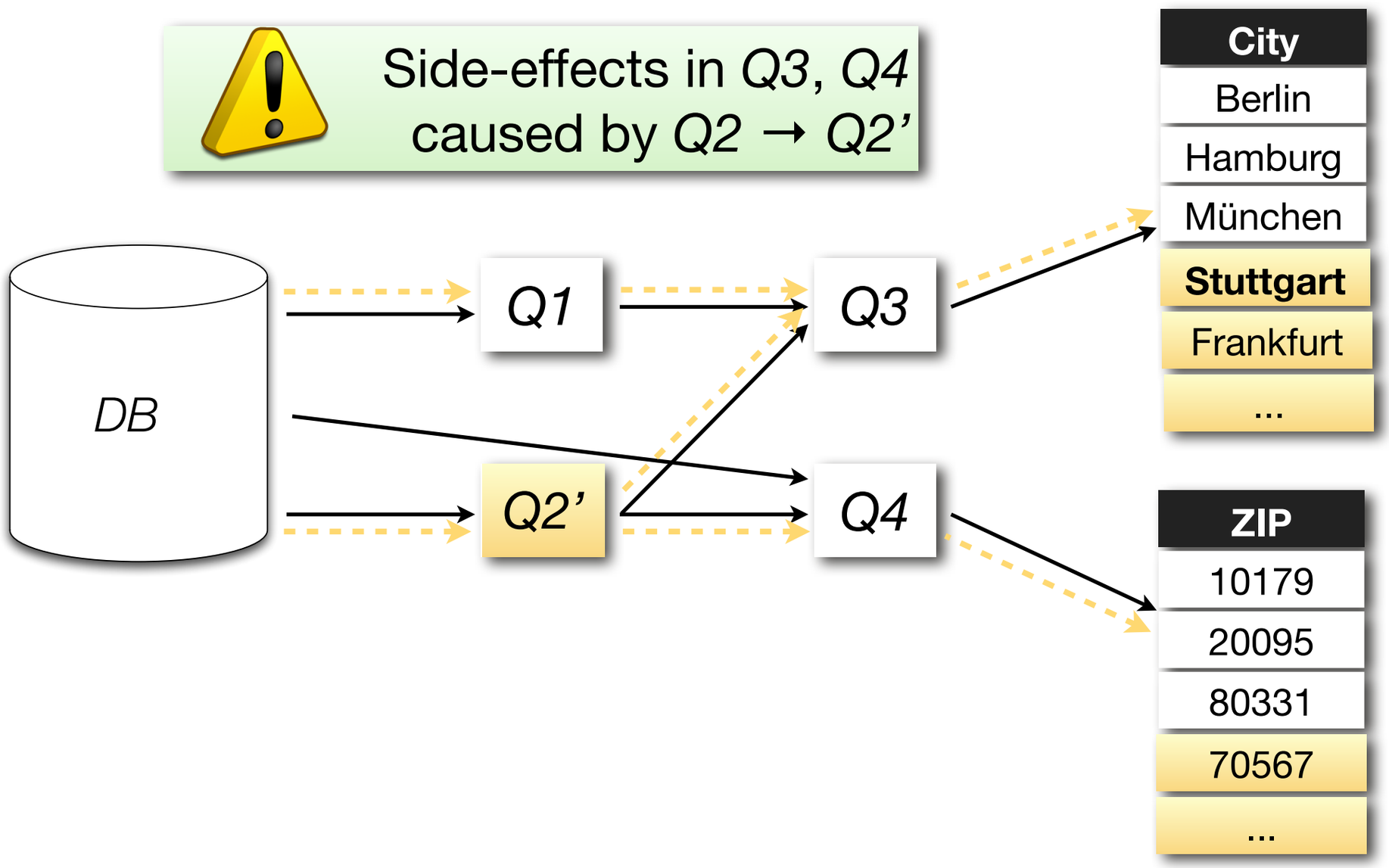


Sample Workflow





Side-effects in *Q3*, *Q4* caused by *Q2* → *Q2'*



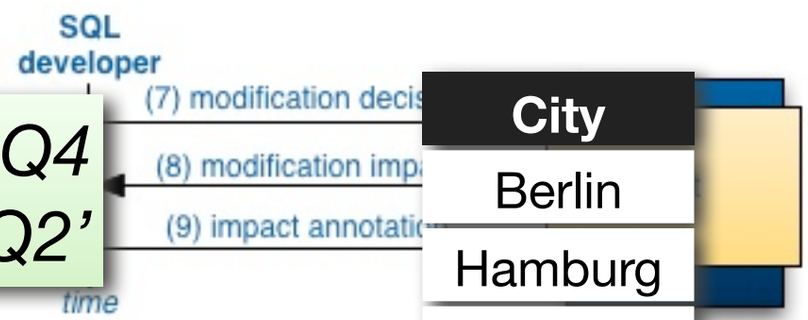
Analyze

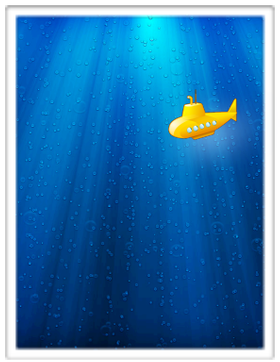
Fix

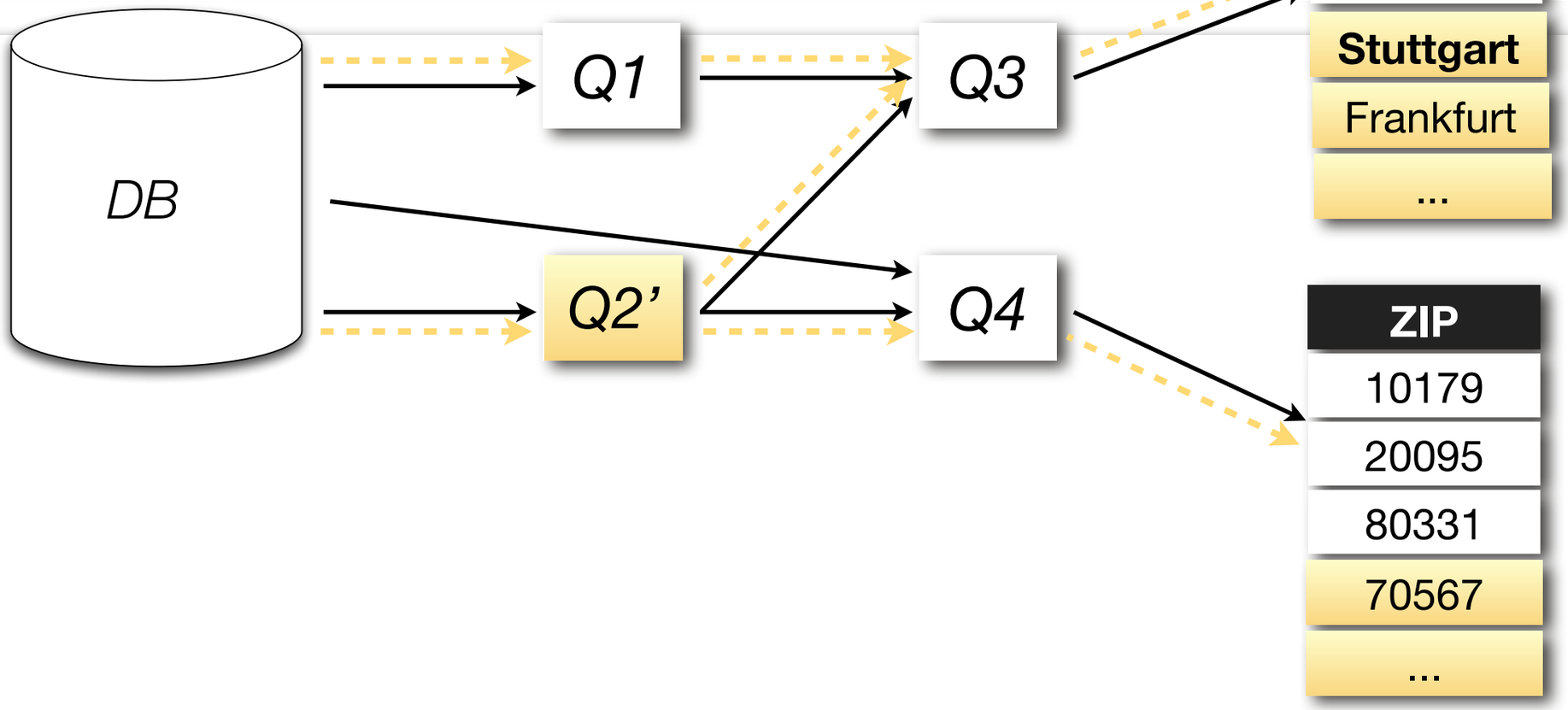
Test

Sampl

Side-effects in *Q3*, *Q4* caused by *Q2* → *Q2'*







Analyze

Fix

Test

modification decision

- submit modification(s) to the query based on the annotations

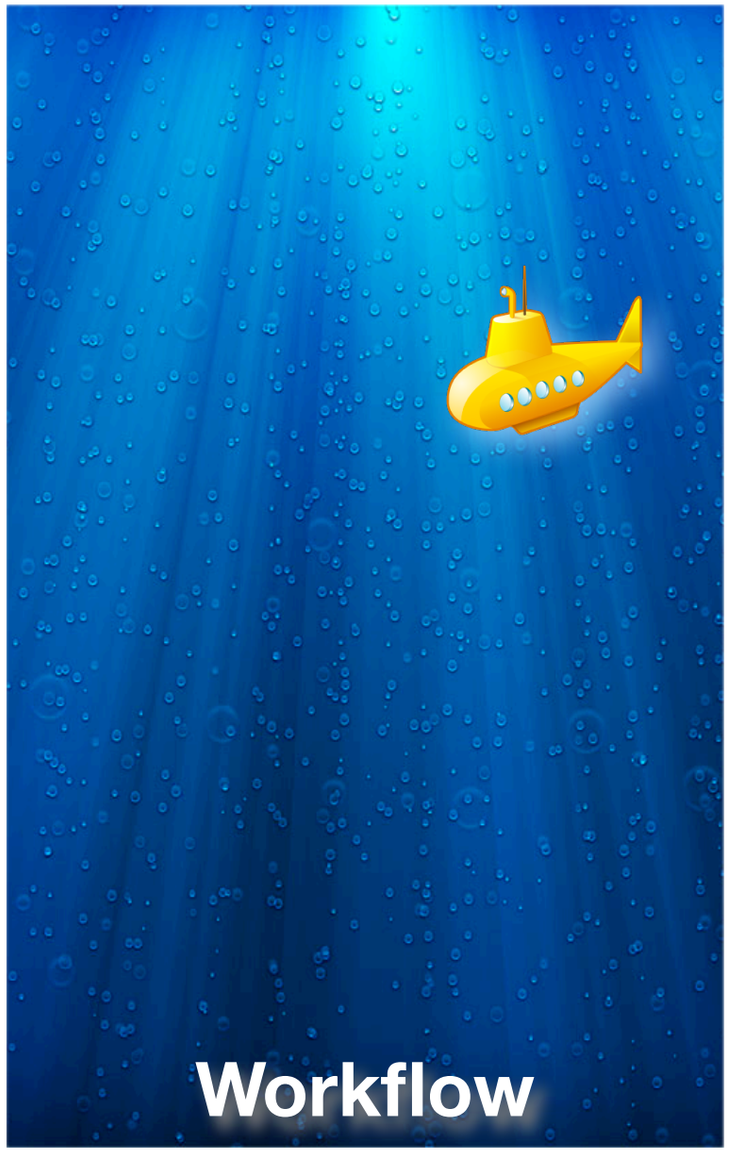
modification impact

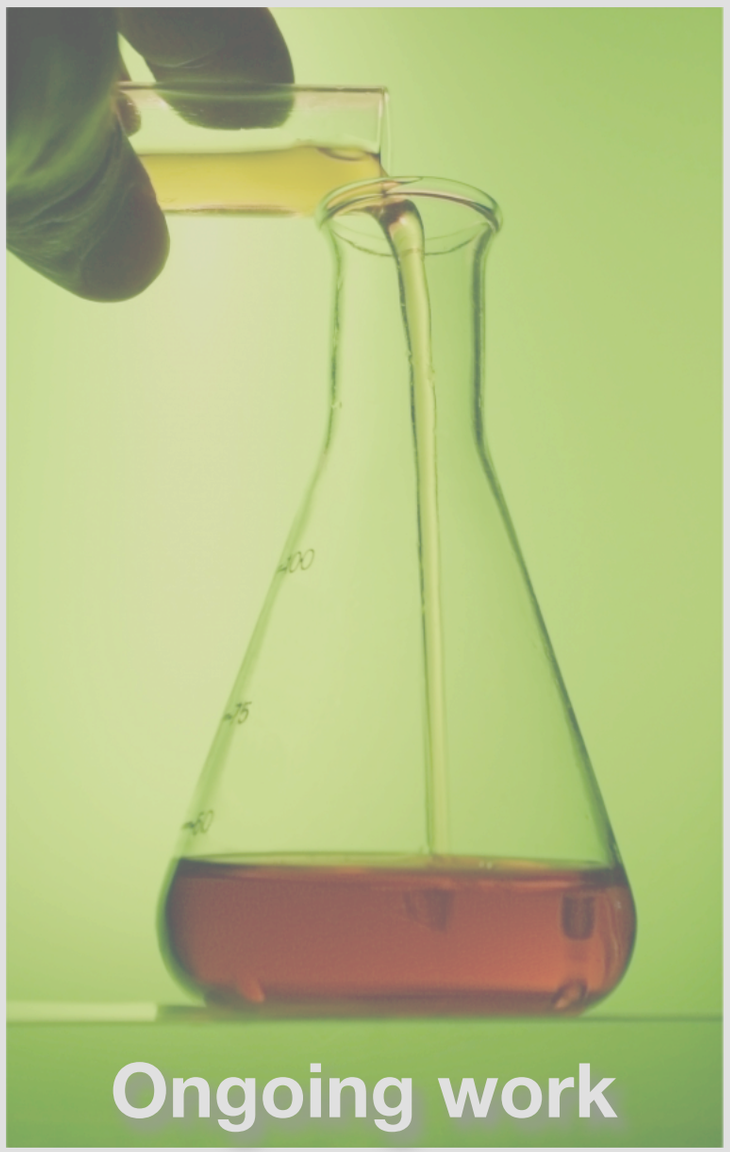
- test the query
- calculating the impact (verifying constraints and report statistics)

impact annotation

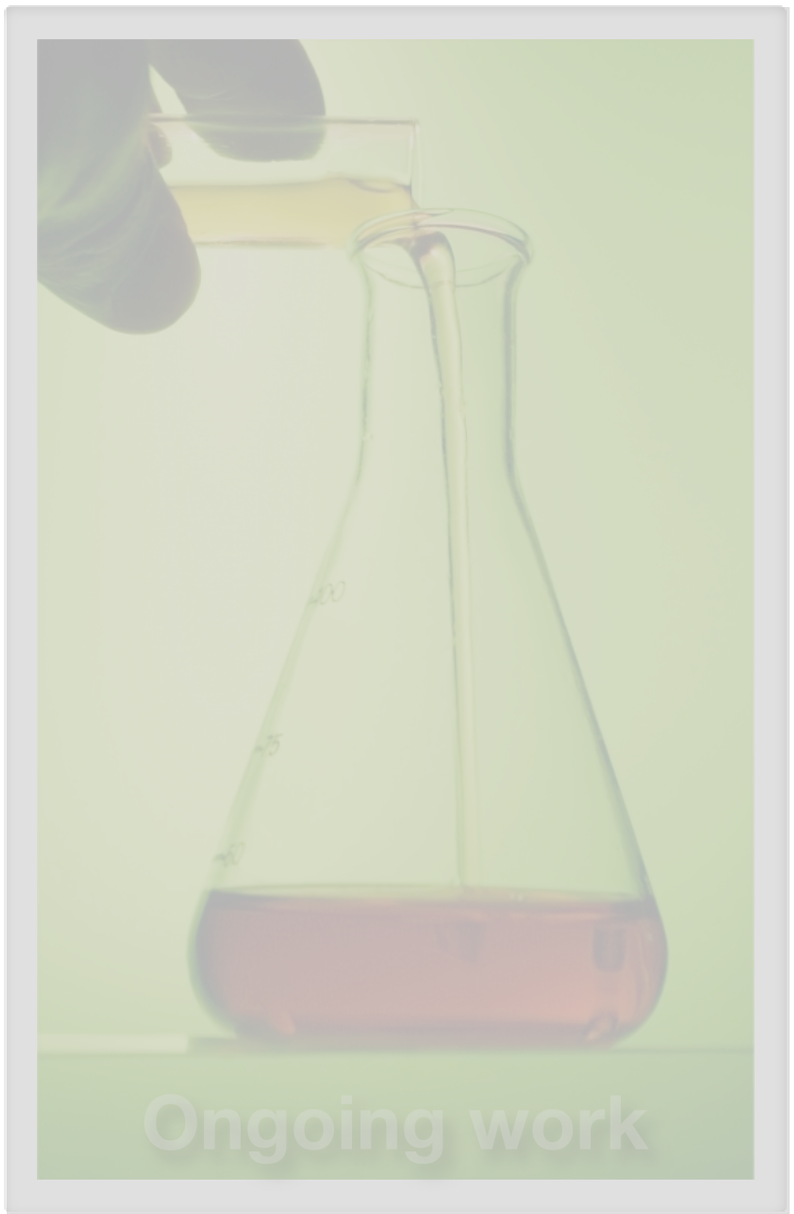
- acceptable changes
- effect on further AFT-cycles and debugging scenario

Agenda

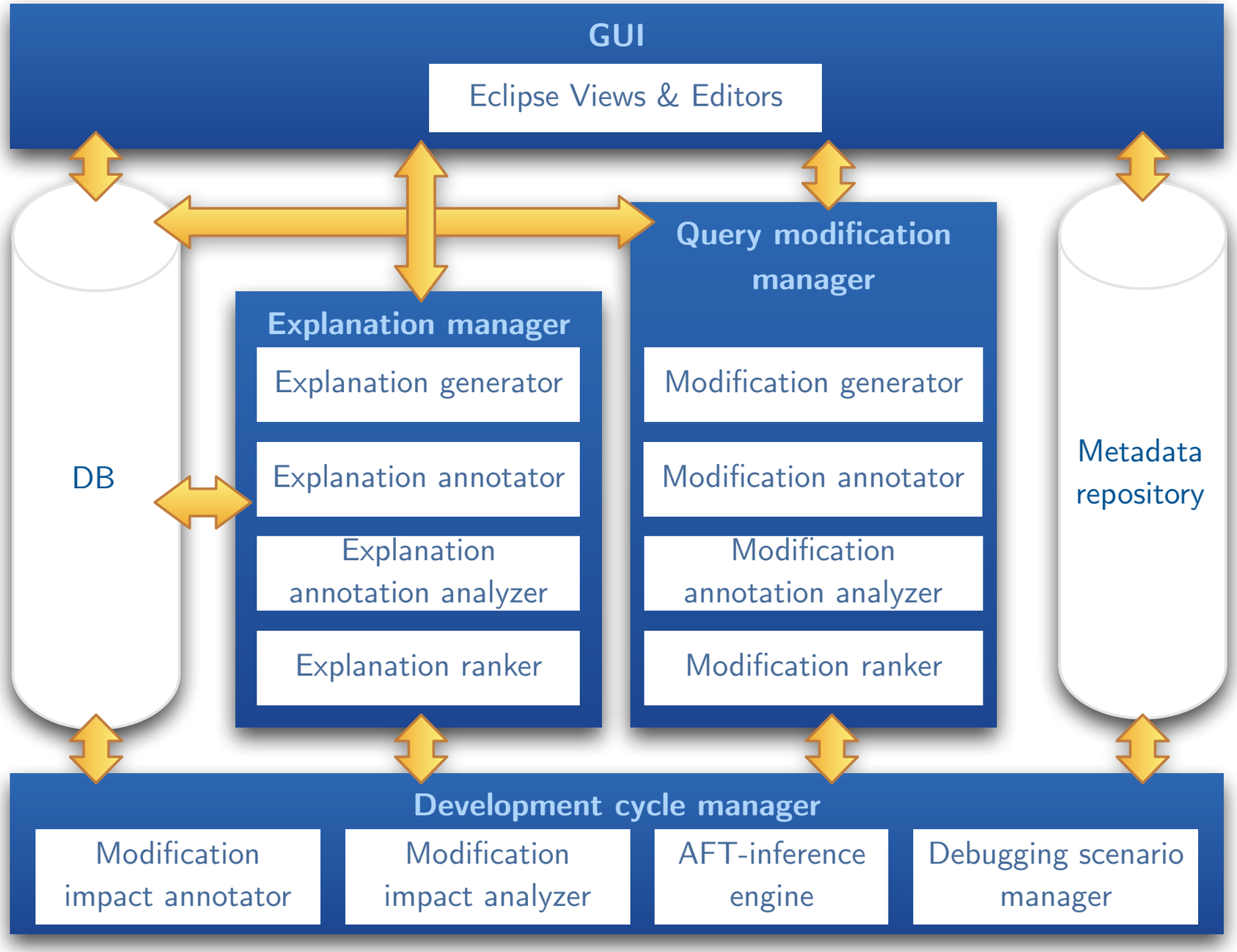




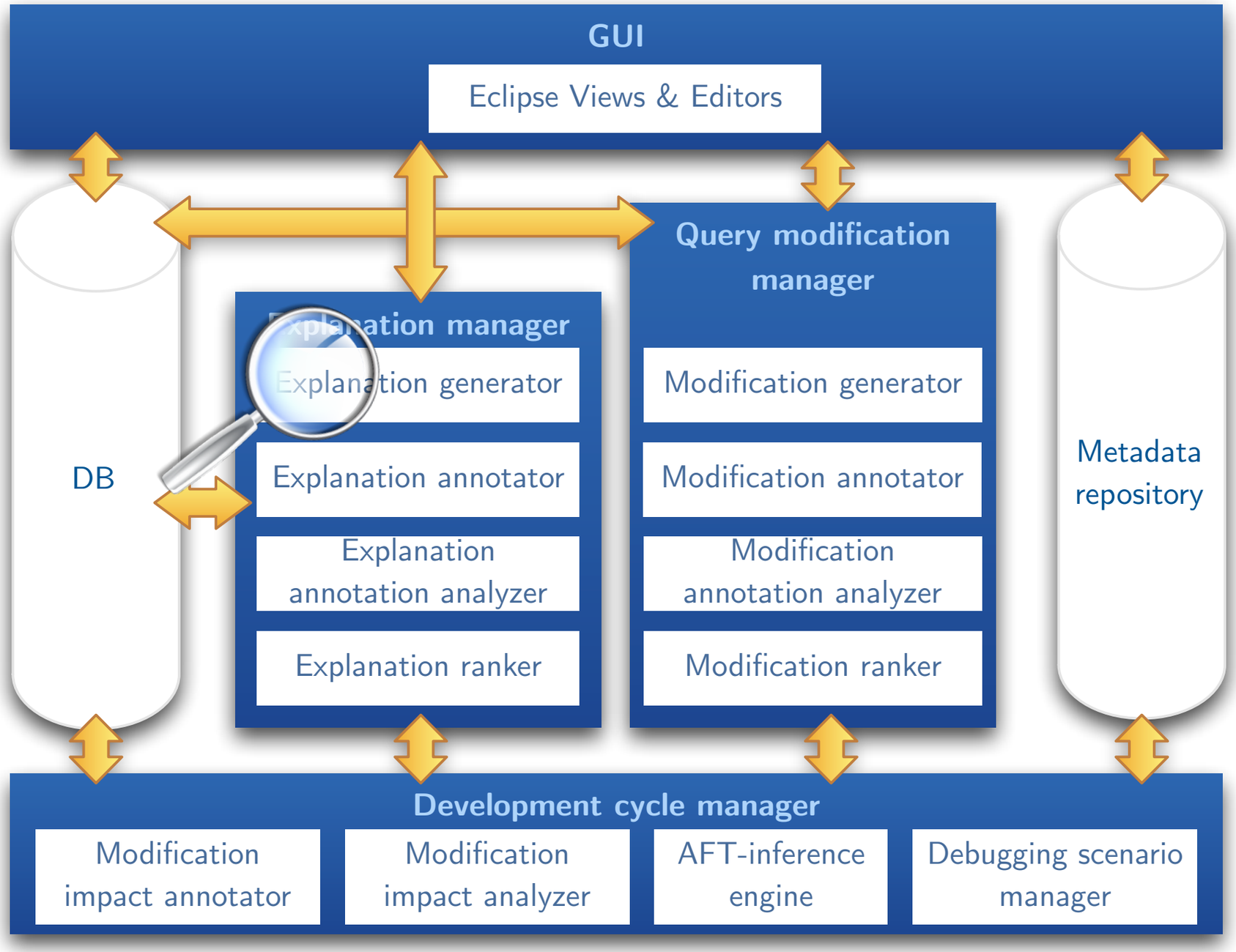
Agenda



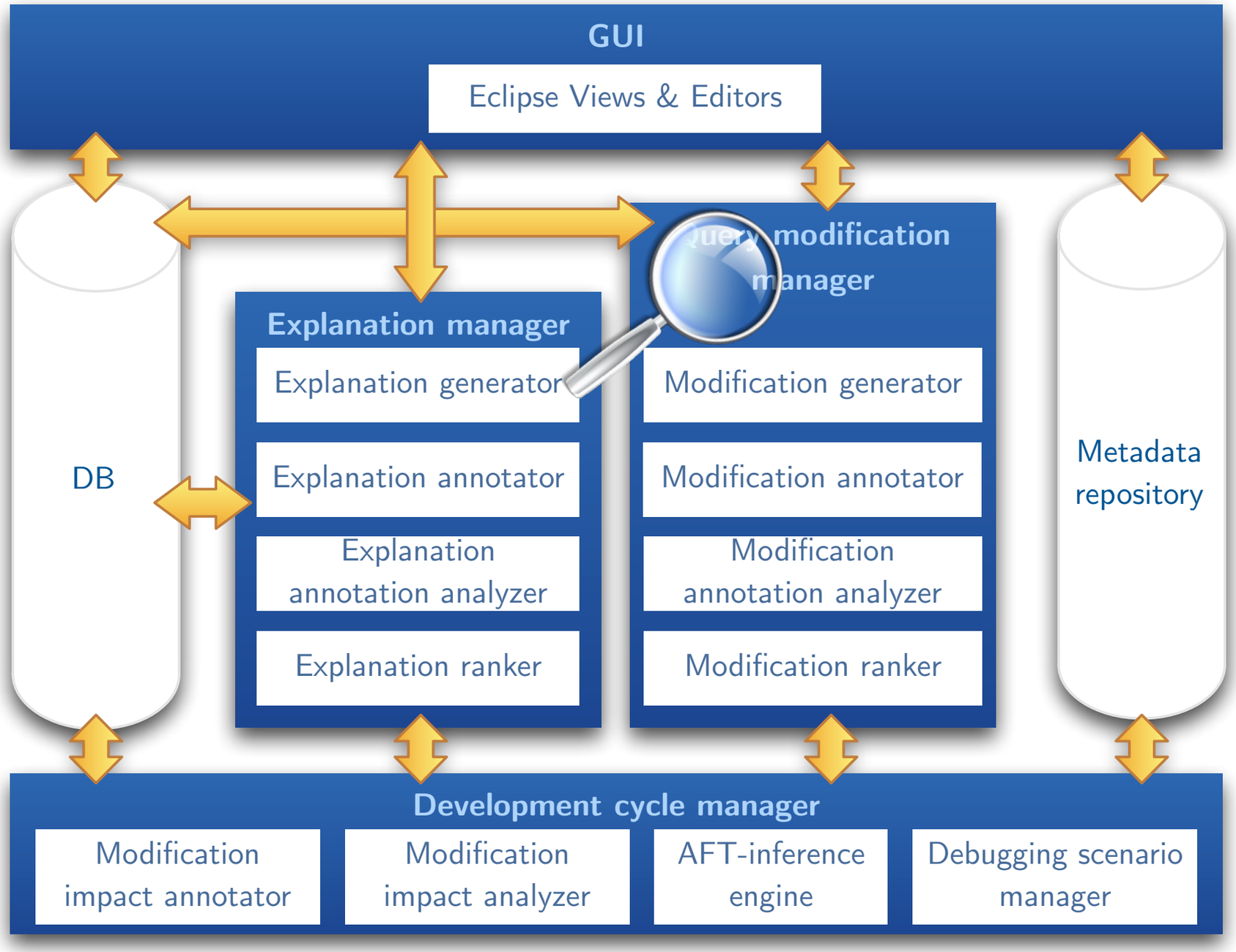
Nautilus Architecture



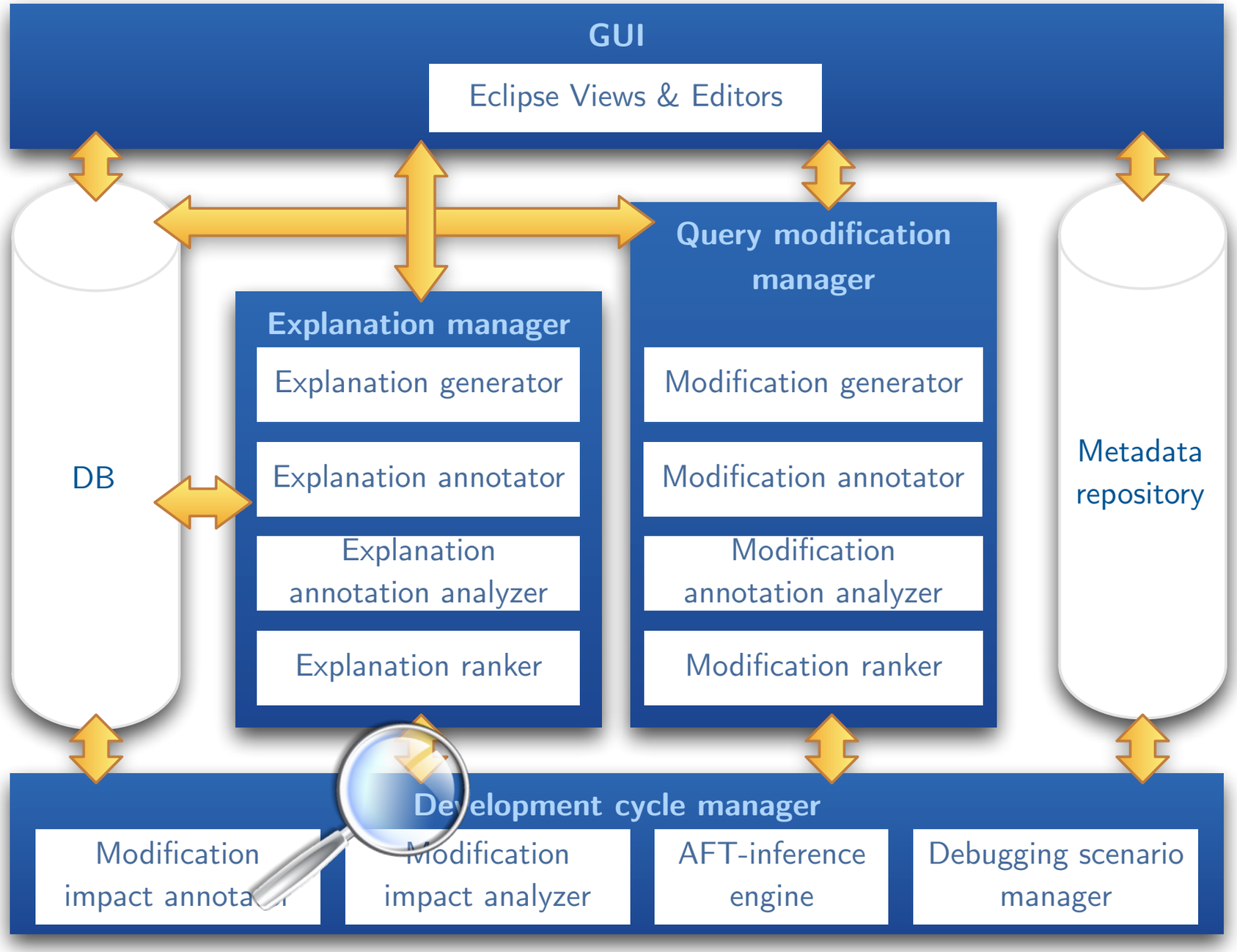
Nautilus Architecture



Nautilus Architecture



Nautilus Architecture



Agenda

Agenda

Conseil - Hybrid explanations

Goal: Ideally pointing out both, the problem of missing source data and the problem of problematic query operators of a non-monotonous query.

Idea:

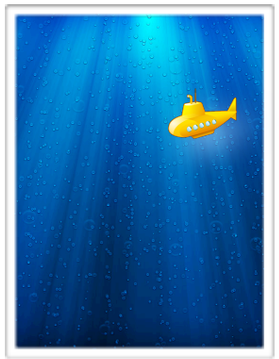
- Conseil follows a positive example to generate explanations.
- A positive example is an existing tuple with a high similarity to the missing answer.
- Annotate the canonical logical tree of the positive example and the missing tuple with passing properties.

Conseil - Idea (continuation)

Idea (continuing):

- Transform the tree to a passing tree, by changing blocking nodes to passing nodes.
- The choice of pointing out the problematic operator or the problematic data tuple(s) is based on a cost model.
- While generating explanations dependencies to previous taken decisions can lead to non optimal global costs (Branch&Bound).

Summary & Outlook



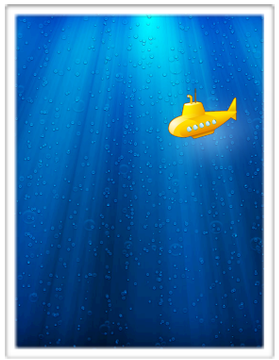
Transformation Lifecycle Management with **Nautilus**

- Semi-automatic tool to support the three phases (AFT) of the currently manual development process
- For various components of the architecture, solutions how to implement the desired functionality were presented.
- A possible benchmark is proposed to evaluate different steps of the AFT cycle.

Outlook

- Evaluate algorithms relevant to different steps with the benchmark.
- Boost the debugging scenario expressiveness and the explanation scope.
- Involve developers through user studies to measure usability.
- Build a real system and evaluate it.

Summary & Outlook



Transformation Lifecycle Management with **Nautilus**

- Semi-automatic tool to support the three phases (AFT) of the currently manual development process
- For various components of the architecture, solutions how to implement the desired functionality were presented.
- A possible benchmark is proposed to evaluate different steps of the AFT cycle.

Outlook

- Evaluate algorithms relevant to different steps with the benchmark.
- Boost the debugging scenario expressiveness and the explanation scope.
- Involve developers through user studies to measure usability.

Nautilus

Thank you for your attention.

http://nautilus-system.org