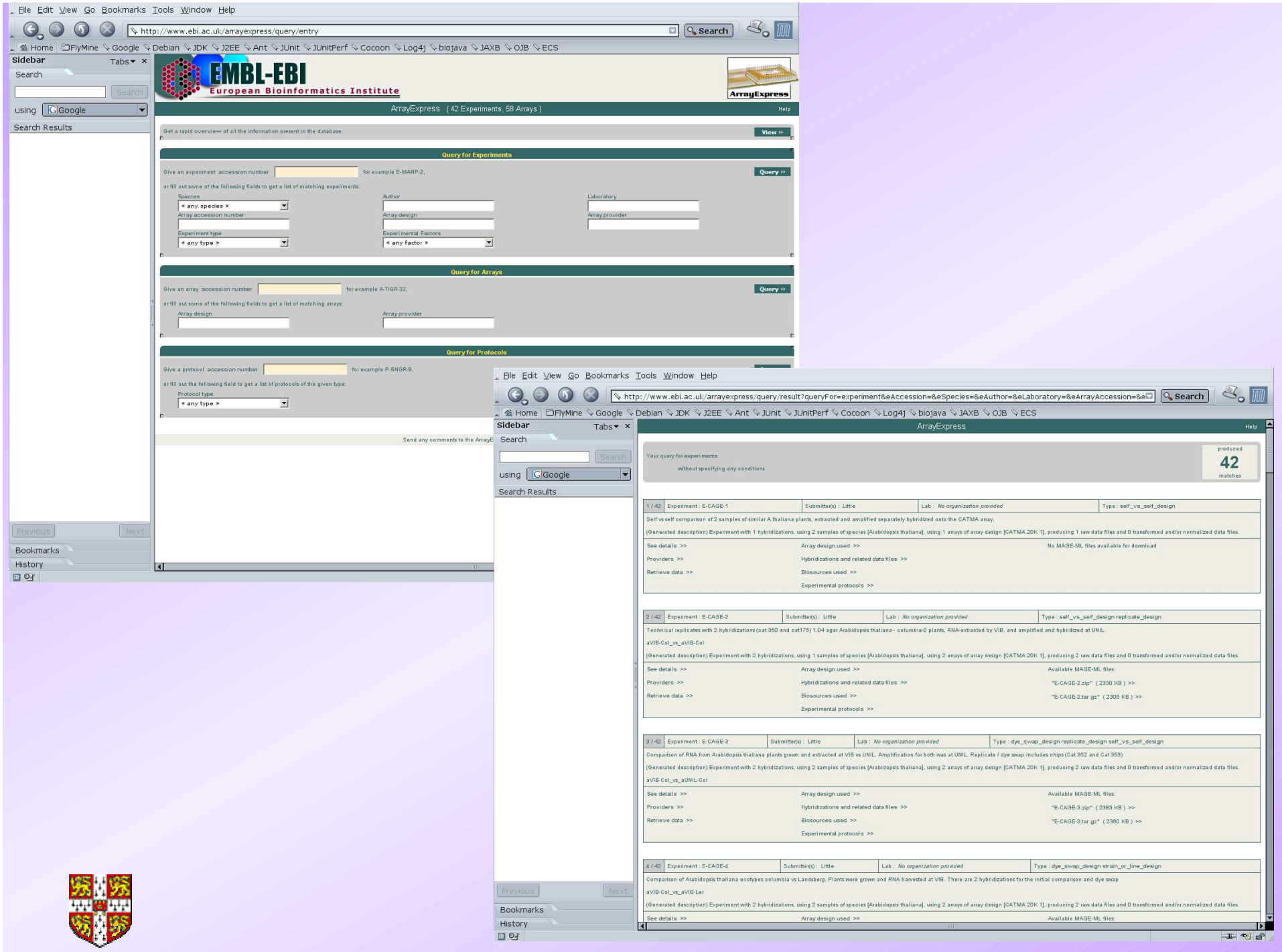
# FlyMine

An integrated database for *Drosophila* and *Anopheles* genomics

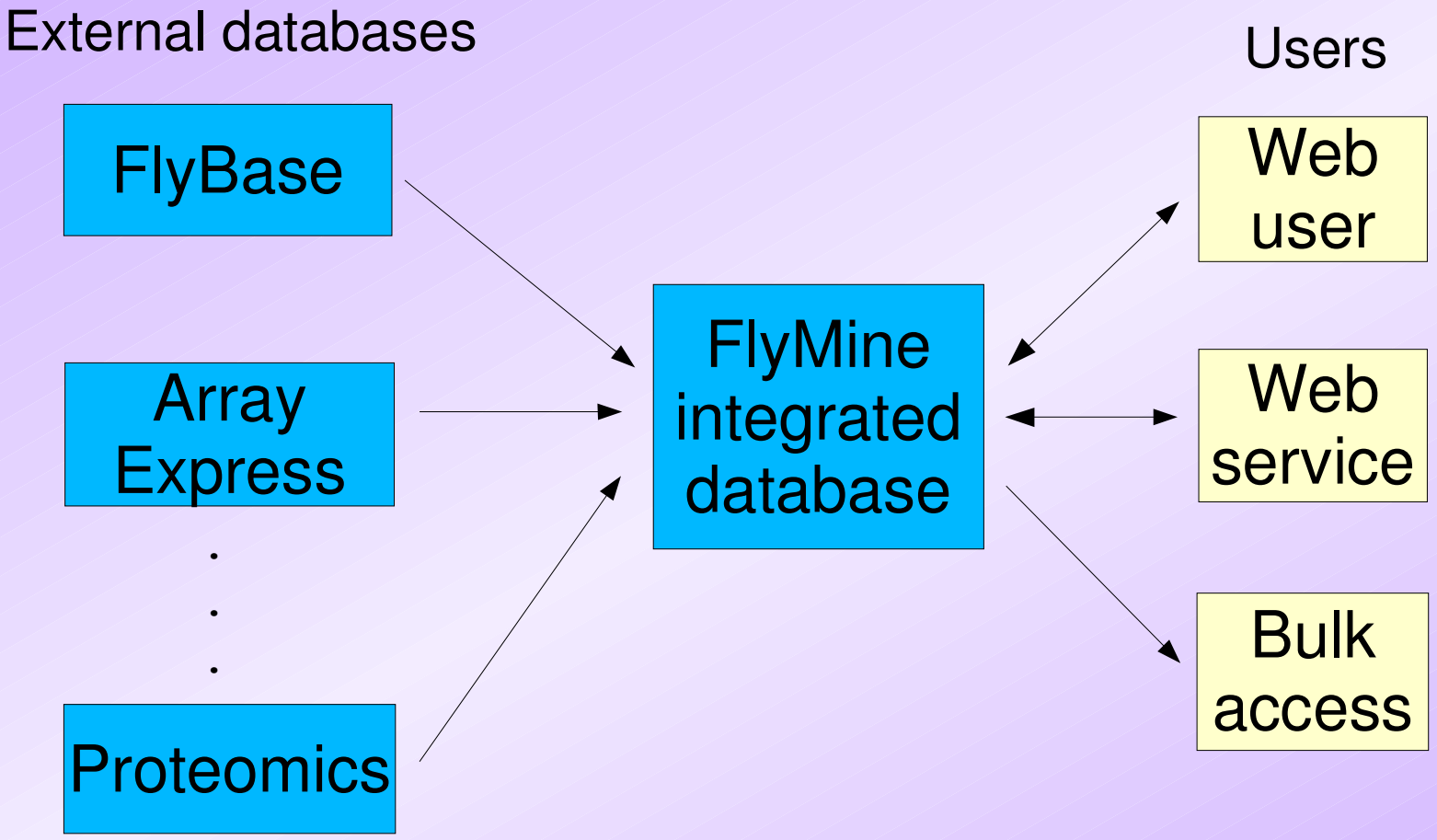
Andrew Varley
24th July 2003

# Why yet another database?

- Currently lots of little databases
  - fine for “browsing”, bad for “querying”
- Hard to query across them
  - lots of “cut and paste” on web pages
- Massive amounts of experimental data (microarray, proteomics) being produced
- Need to tie this information together


File Edit View Go Bookmarks Tools Window Help
http://www.ebi.ac.uk/arrayexpress/query/entry
Search
Home FlyMine Google Debian JDK J2EE Ant JUnit JUnitPerf Cocoon Log4j biojava JAXB OJB ECS
Sidebar
Tabs
Search
using Google
Search Results
Previous
Next
Bookmarks
History
EMBL-EBI
European Bioinformatics Institute
ArrayExpress
ArrayExpress ( 42 Experiments, 58 Arrays )
Help
Get a rapid overview of all the information present in the database.
View »
Query for Experiments
Give an experiment accession number
for example E-MANP-2,
Query »
or fill out some of the following fields to get a list of matching experiments:
Species
« any species »
Author
Laboratory
Array accession number
Array design
Array provider
Experiment type
« any type »
Experimental Factors
« any factor »
Query for Arrays
Give an array accession number
for example A-TIGR-32,
Query »
or fill out some of the following fields to get a list of matching arrays:
Array design
Array provider
Query for Protocols
Give a protocol accession number
for example P-SNGR-8,
or fill out the following field to get a list of protocols of the given type:
Protocol type
« any type »
Send any comments to the ArrayE
http://www.ebi.ac.uk/arrayexpress/query/result?queryFor=experiment&eAccession=&eSpecies=&eAuthor=&eLaboratory=&eArrayAccession=&e
Your query for experiments
without specifying any conditions
produced
42
matches
1 / 42 Experiment : E-CAGE-1 Submitter(s) : Little Lab : No organization provided Type : self_vs_self_design
Self vs self comparison of 2 samples of similar A.thaliana plants, extracted and amplified separately hybridized onto the CATMA array.
(Generated description) Experiment with 1 hybridizations, using 2 samples of species [Arabidopsis thaliana], using 1 arrays of array design [CATMA 20K 1], producing 1 raw data files and 0 transformed and/or normalized data files.
See details >>
Array design used >>
No MAGE-ML files available for download
Providers >>
Hybridizations and related data files >>
Retrieve data >>
Biosources used >>
Experimental protocols >>
2 / 42 Experiment : E-CAGE-2 Submitter(s) : Little Lab : No organization provided Type : self_vs_self_design replicate_design
Technical replicates with 2 hybridizations (cat 350 and cat175) 1.04 agar Arabidopsis thaliana - columbia-0 plants, RNA-extracted by VIB, and amplified and hybridized at UNIL.
aVIB-Col_vs_aVIB-Col
(Generated description) Experiment with 2 hybridizations, using 1 samples of species [Arabidopsis thaliana], using 2 arrays of array design [CATMA 20K 1], producing 2 raw data files and 0 transformed and/or normalized data files.
See details >>
Array design used >>
Available MAGE-ML files:
Providers >>
Hybridizations and related data files >>
"E-CAGE-2.zip" ( 2330 KB ) >>
Retrieve data >>
Biosources used >>
"E-CAGE-2.tar.gz" ( 2305 KB ) >>
Experimental protocols >>
3 / 42 Experiment : E-CAGE-3 Submitter(s) : Little Lab : No organization provided Type : dye_swap_design replicate_design self_vs_self_design
Comparison of RNA from Arabidopsis thaliana plants grown and extracted at VIB vs UNIL. Amplification for both was at UNIL. Replicate / dye swap includes chips (Cat 352 and Cat 353)
(Generated description) Experiment with 2 hybridizations, using 2 samples of species [Arabidopsis thaliana], using 2 arrays of array design [CATMA 20K 1], producing 2 raw data files and 0 transformed and/or normalized data files.
aVIB-Col_vs_aUNIL-Col
See details >>
Array design used >>
Available MAGE-ML files:
Providers >>
Hybridizations and related data files >>
"E-CAGE-3.zip" ( 2383 KB ) >>
Retrieve data >>
Biosources used >>
"E-CAGE-3.tar.gz" ( 2380 KB ) >>
Experimental protocols >>
4 / 42 Experiment : E-CAGE-4 Submitter(s) : Little Lab : No organization provided Type : dye_swap_design strain_or_line_design
Comparison of Arabidopsis thaliana ecotypes columbia vs Landsberg. Plants were grown and RNA harvested at VIB. There are 2 hybridizations for the initial comparison and dye swap.
aVIB-Col_vs_aVIB-Ler
(Generated description) Experiment with 2 hybridizations, using 2 samples of species [Arabidopsis thaliana], using 2 arrays of array design [CATMA 20K 1], producing 2 raw data files and 0 transformed and/or normalized data files.
See details >>
Array design used >>
Available MAGE-ML files:

# Outline

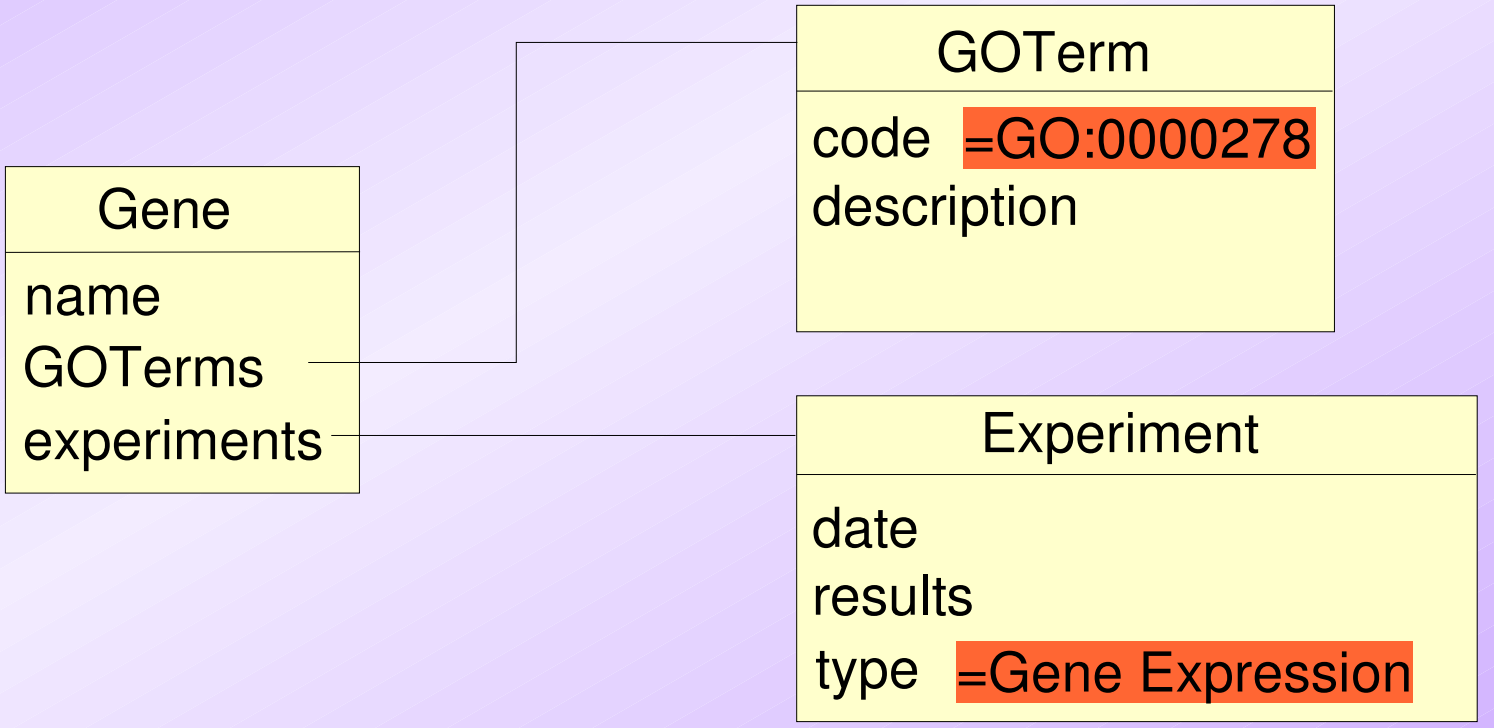

External databases
FlyBase
Array Express
Proteomics
FlyMine integrated database
Users
Web user
Web service
Bulk access

# Aims

- Integrate data from multiple sources
- Allow arbitrary queries from users
- Queries based on objects, not SQL
- Multiple query interfaces (Java, OQL, graphical, etc)
- Different classes of user (web-based, SOAP)
- Open source!

# Query interfaces – graphical

*"Show gene expression data for genes which have GO term GO:0000278 applied"*




# Query interfaces – OQL

*"Show gene expression data for genes which have GO term GO:0000278 applied"*

```
SELECT gene, exp
FROM Gene AS gene, GOTerm AS term,
     Experiment AS exp
WHERE gene.GOTerms CONTAINS term
AND term.code = "GO:0000278"
AND gene.experiments CONTAINS exp
AND exp.type = "Gene Expression"
```

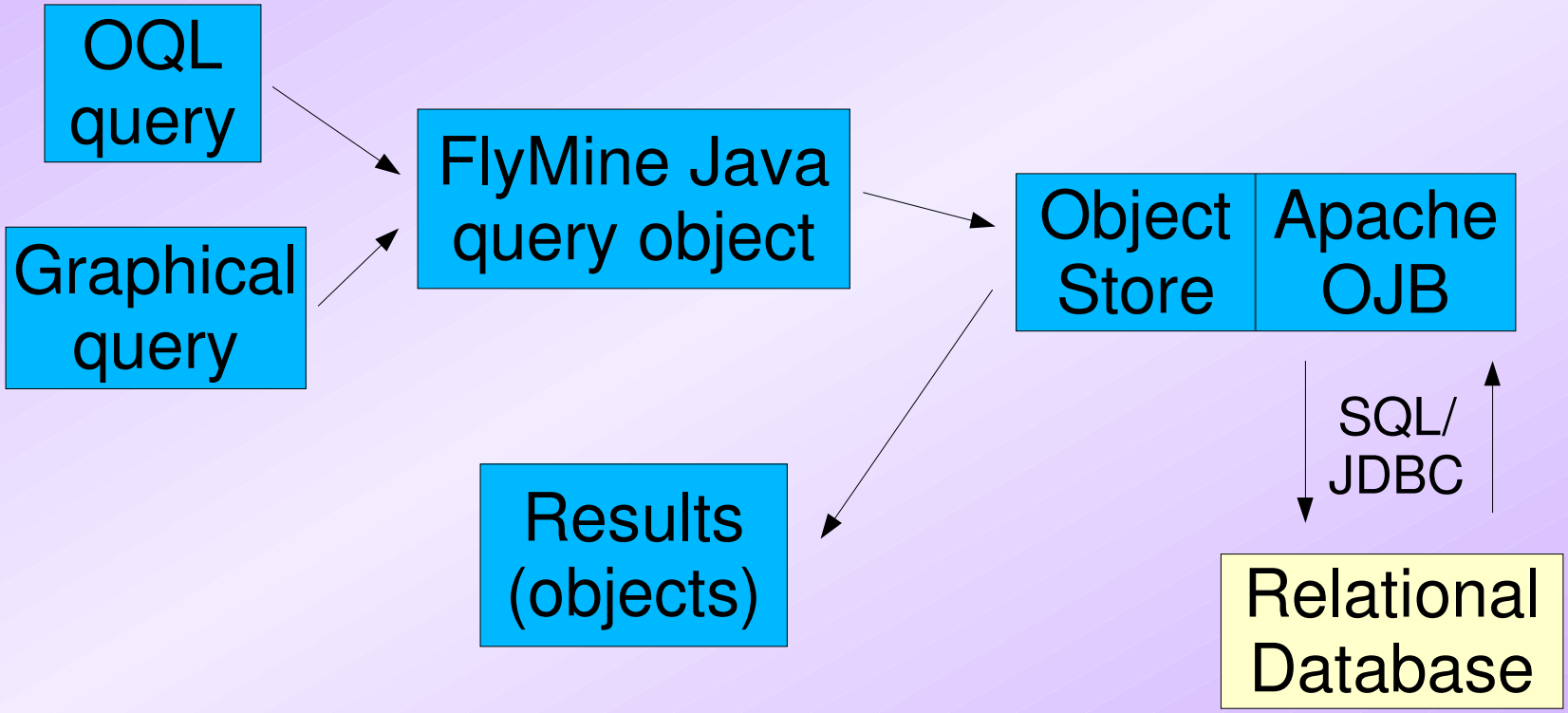
# Query interfaces – Java

*"Show gene expression data for genes which have GO term GO:0000278 applied"*

```
Query q = new Query();
QueryClass qcGene = new QueryClass(Gene.class);
QueryClass qcTerm = new QueryClass(GOTerm.class);
QueryField qfGeneTerm = new QueryField(qcGene, "GOTerms")
ConstraintSet cs = new ConstraintSet();
Constraint con1 = new ContainsConstraint(qfGeneTerm,
                  ContainsConstraint.CONTAINS, qcTerm);
cs.add(con1);
.
.
q.setConstraint(cs);
```

# FlyMine ObjectStore

- Based on heavily modified Apache OJB

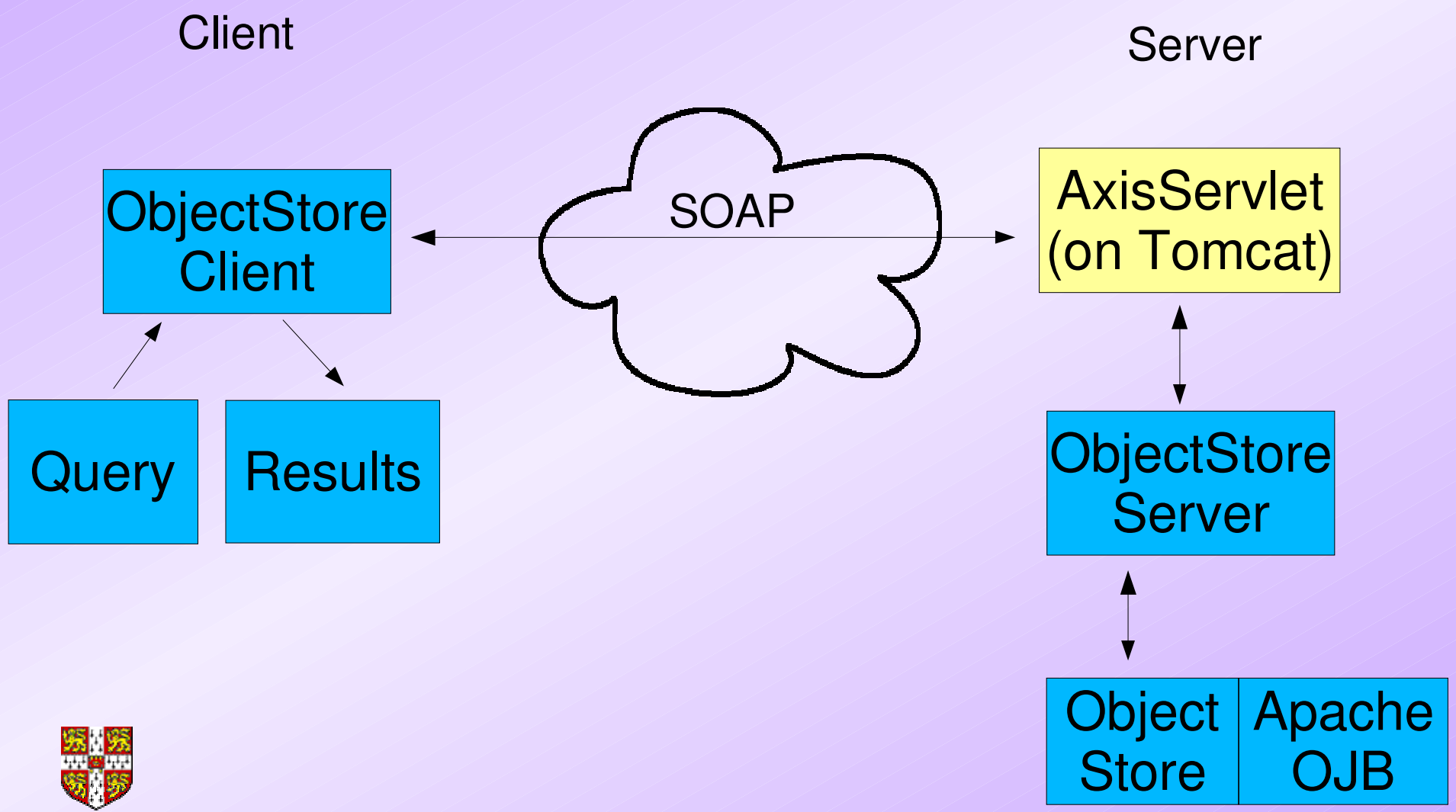



# ObjectStore interface

```
public interface ObjectStore {

  Results execute(Query q);
  List execute(Query q, int start, int limit);
  int count(Query q);
  ExplainResult estimate(Query q);
  Object getObjectByExample(Object obj);
  Model getModel();
}
```

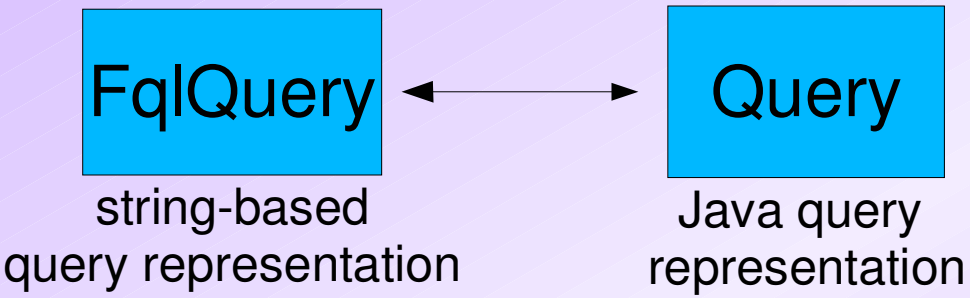
# Web service


Client
Server
ObjectStore Client
SOAP
AxisServlet (on Tomcat)
Query
Results
ObjectStore Server
Object Store
Apache OJB

# Web service

- Objects to encode
  - data objects (all are beans)
  - Query
  - ResultsRow (a list)
  - ExplainResult (a few integers)
  - Model (already have XML format)
  - object proxies

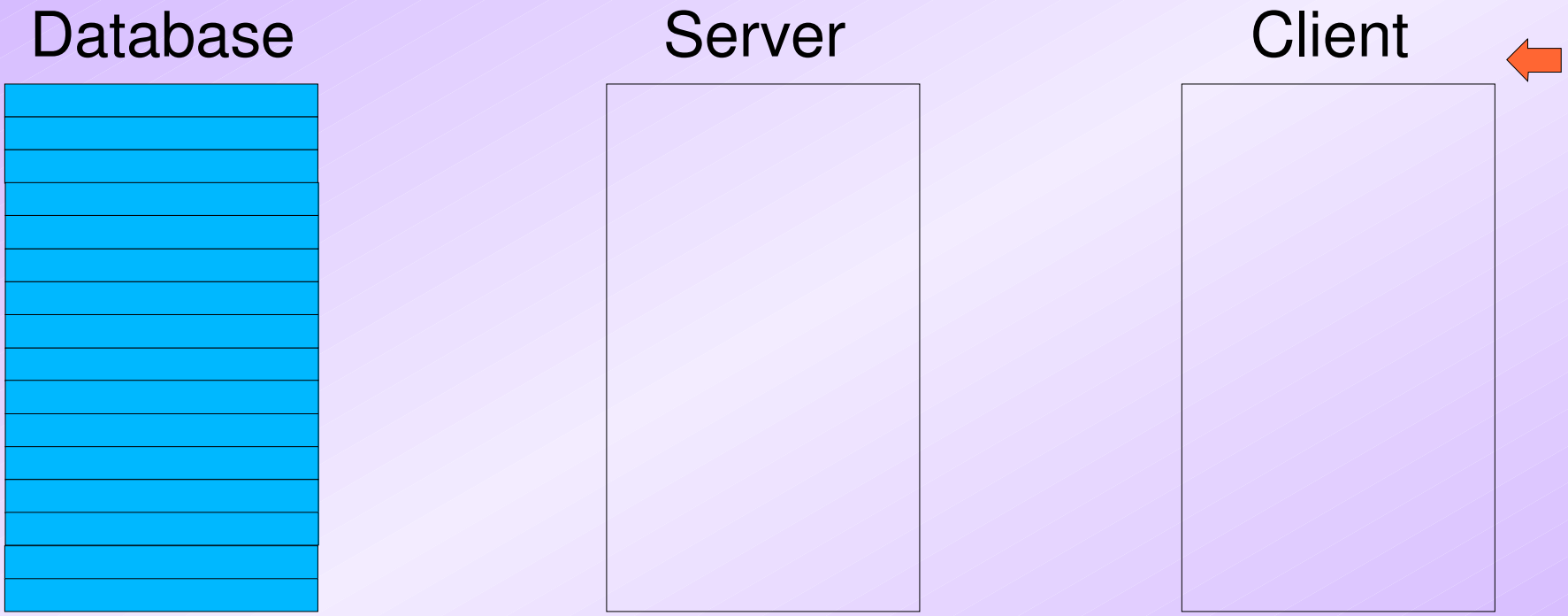
# ObjectStoreServer



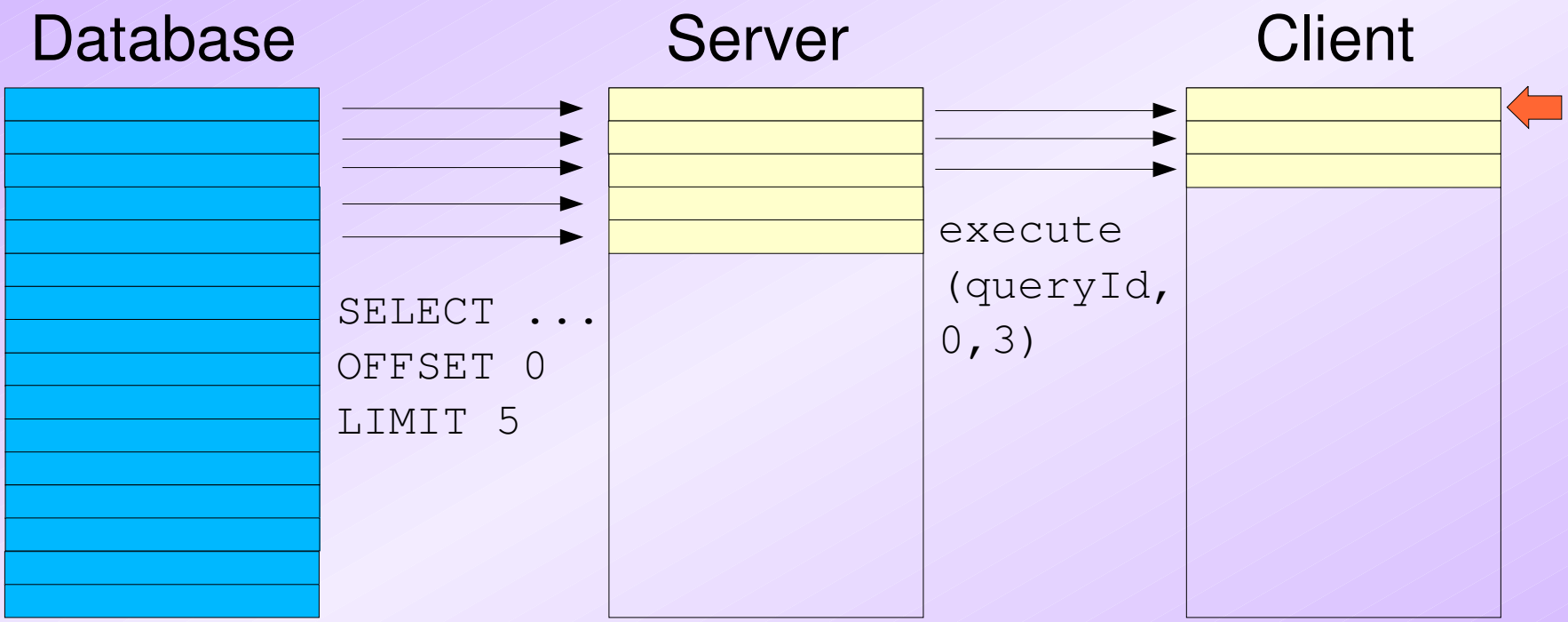

```
public class ObjectStoreServer {
   int registerQuery(FqlQuery q);
   List execute(int queryId, int start, int end);
   int count(int queryId);
   ExplainResult estimate(int queryId);
   Object getObjectByExample(Object obj);
   Model getModel();
}
```

```
 <soapenv:Body>
  <registerQuery
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <arg0 href="#id0"/>
  </registerQuery>
  <multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="fql-query"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
   <packageName
xsi:type="xsd:string">org.flymine.model.tutorial</packageName>
   <queryString xsi:type="xsd:string">select g from Gene as
g</queryString>
  </multiRef>
 </soapenv:Body>
```

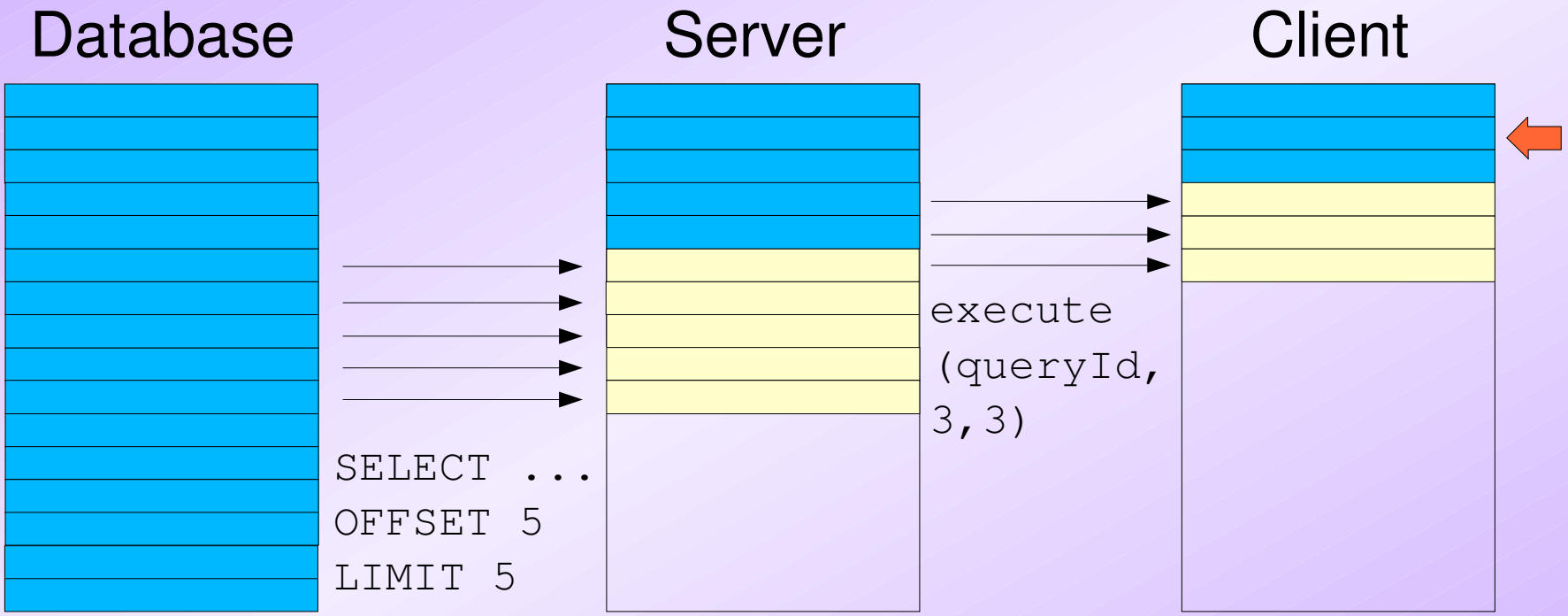
# Results



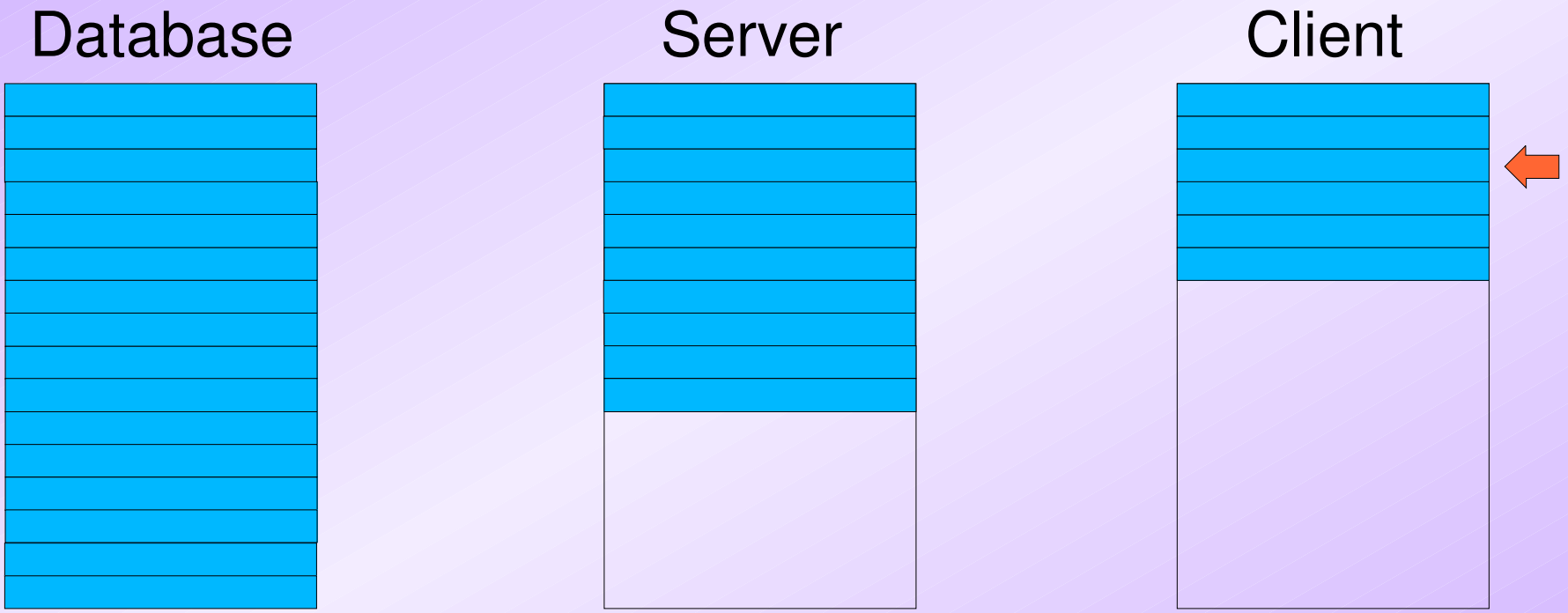

# Results


Database
Server
Client
SELECT ...
OFFSET 0
LIMIT 5
execute
(queryId,
0,3)

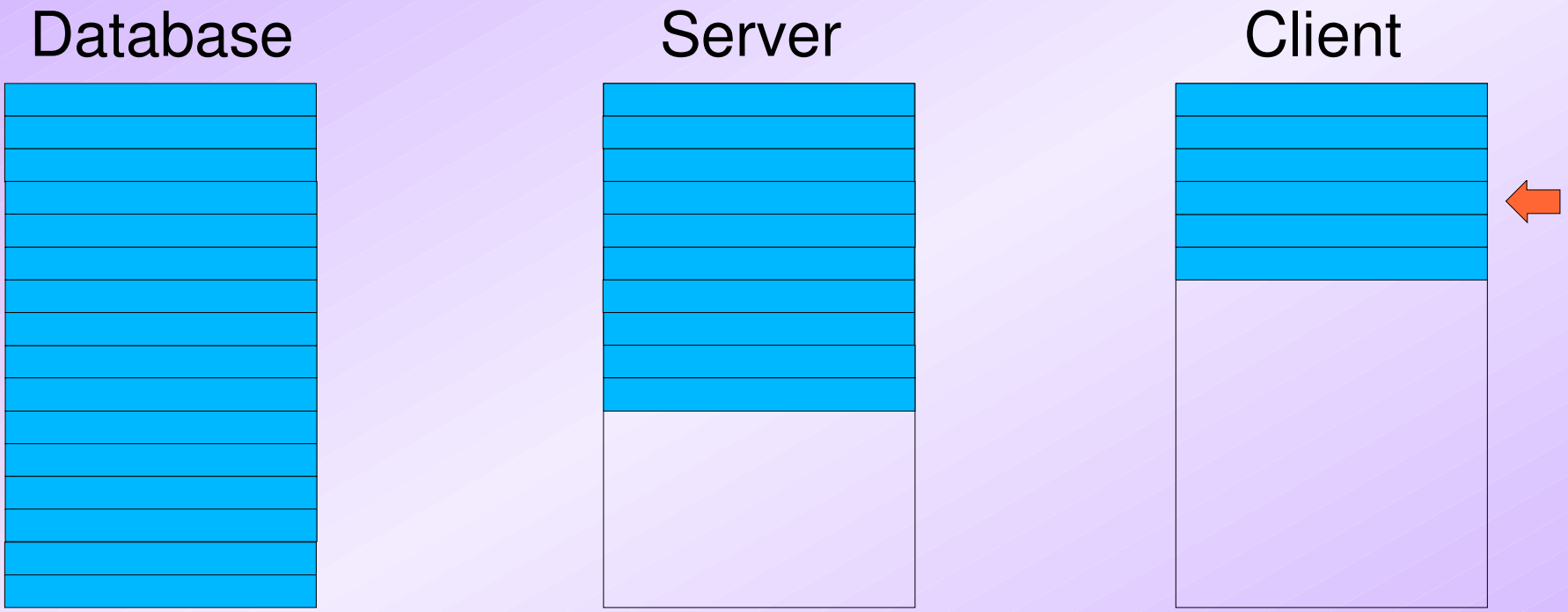
# Results


Database
Server
Client
SELECT ...
OFFSET 5
LIMIT 5
execute
(queryId,
3,3)

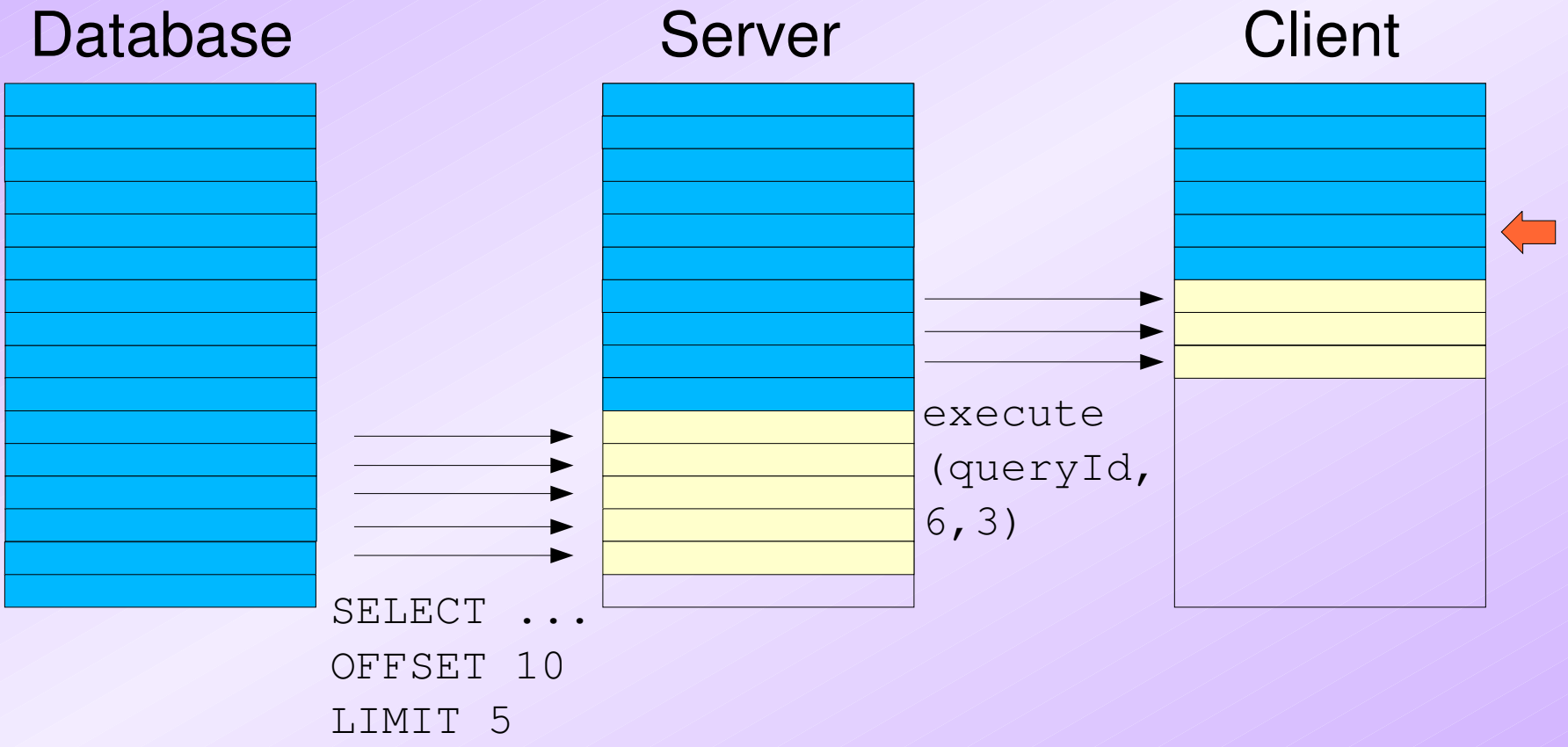
# Results




# Results

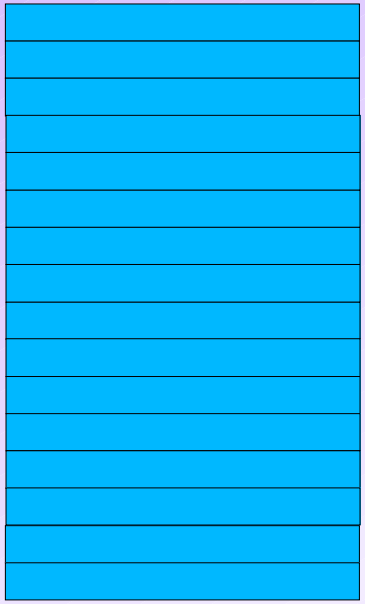



# Results

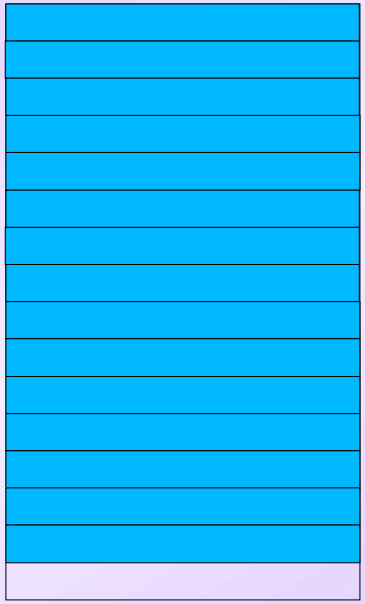

Database
Server
Client
execute
(queryId,
6,3)
SELECT ...
OFFSET 10
LIMIT 5

# Results

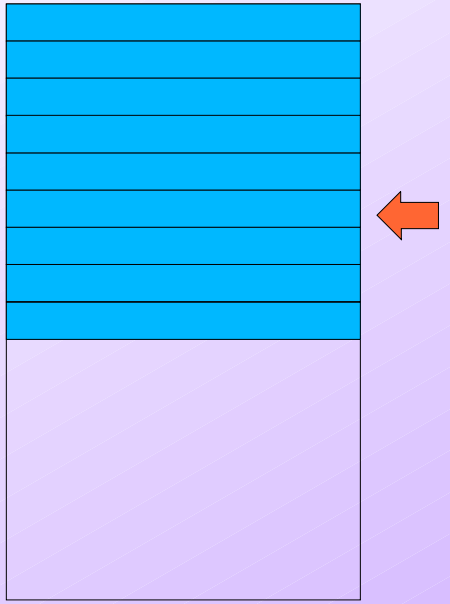
Database



Server

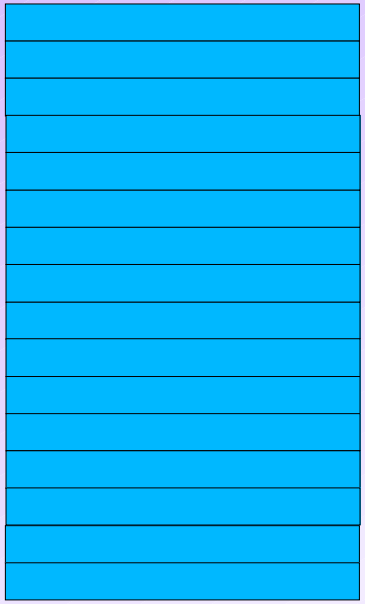


Client

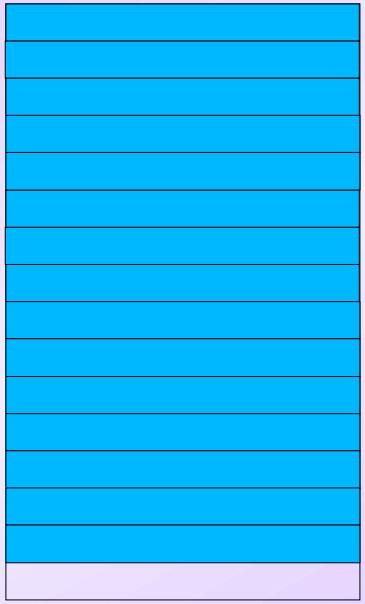


# Results

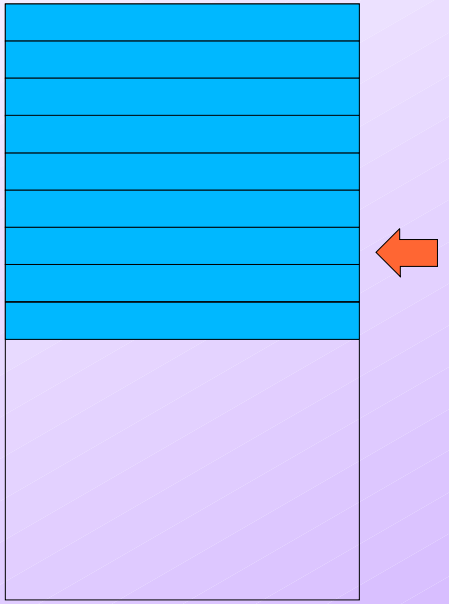
Database



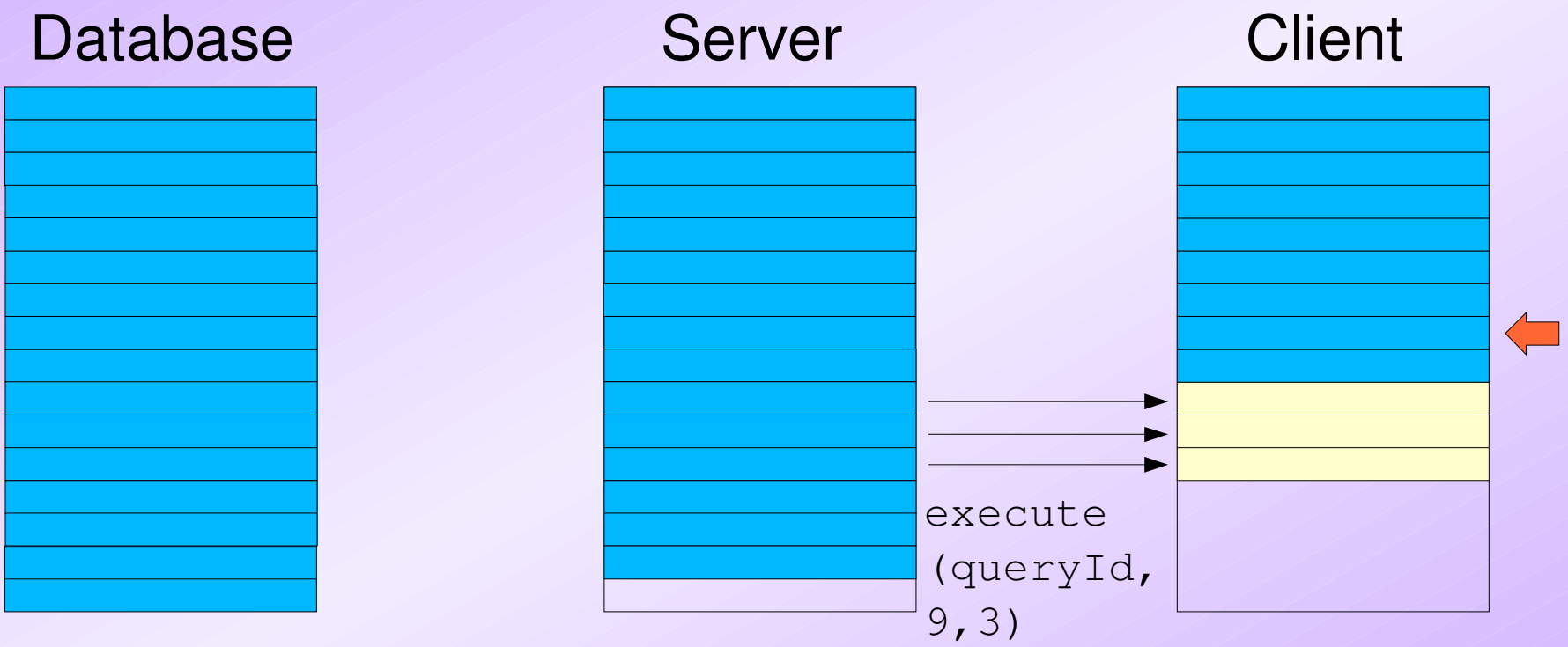
Server



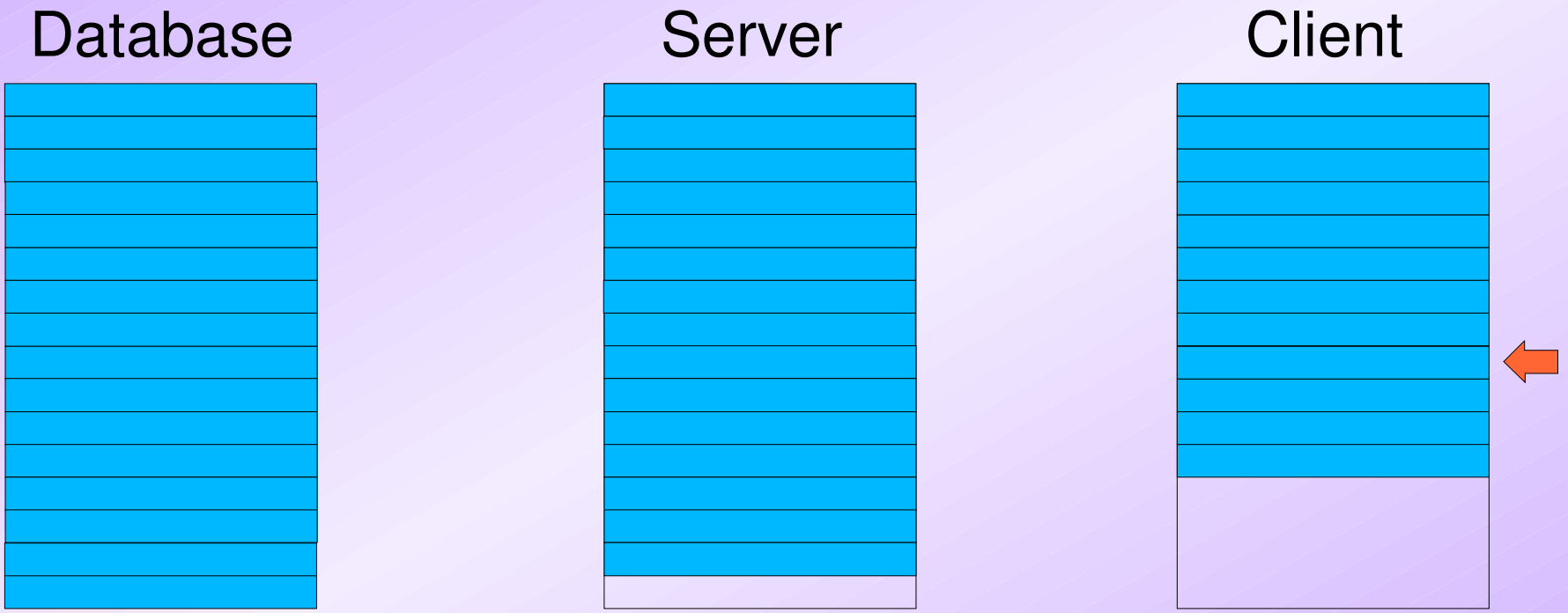
Client



# Results

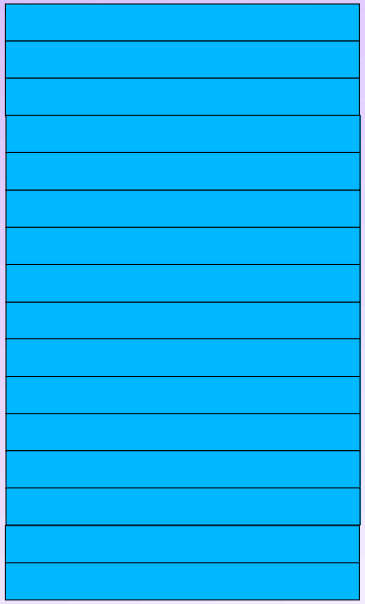



# Results

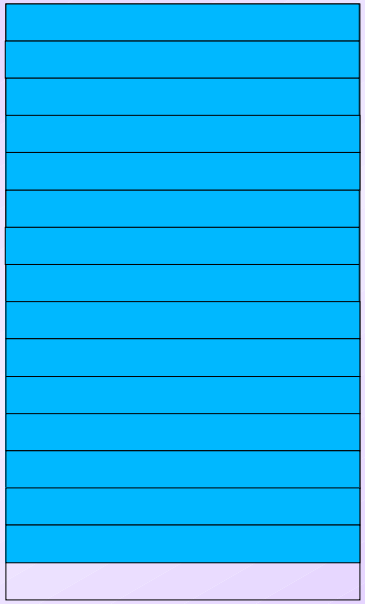



# Results

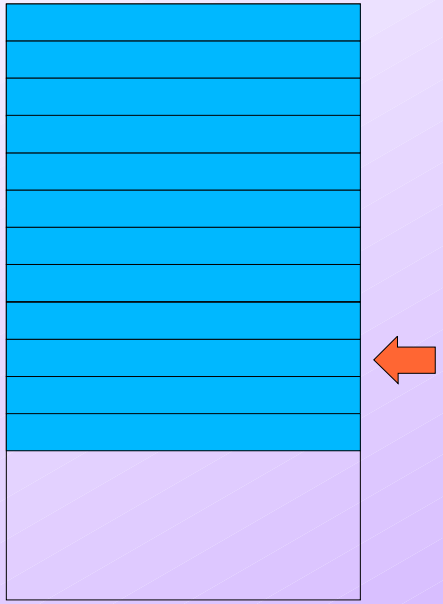
Database



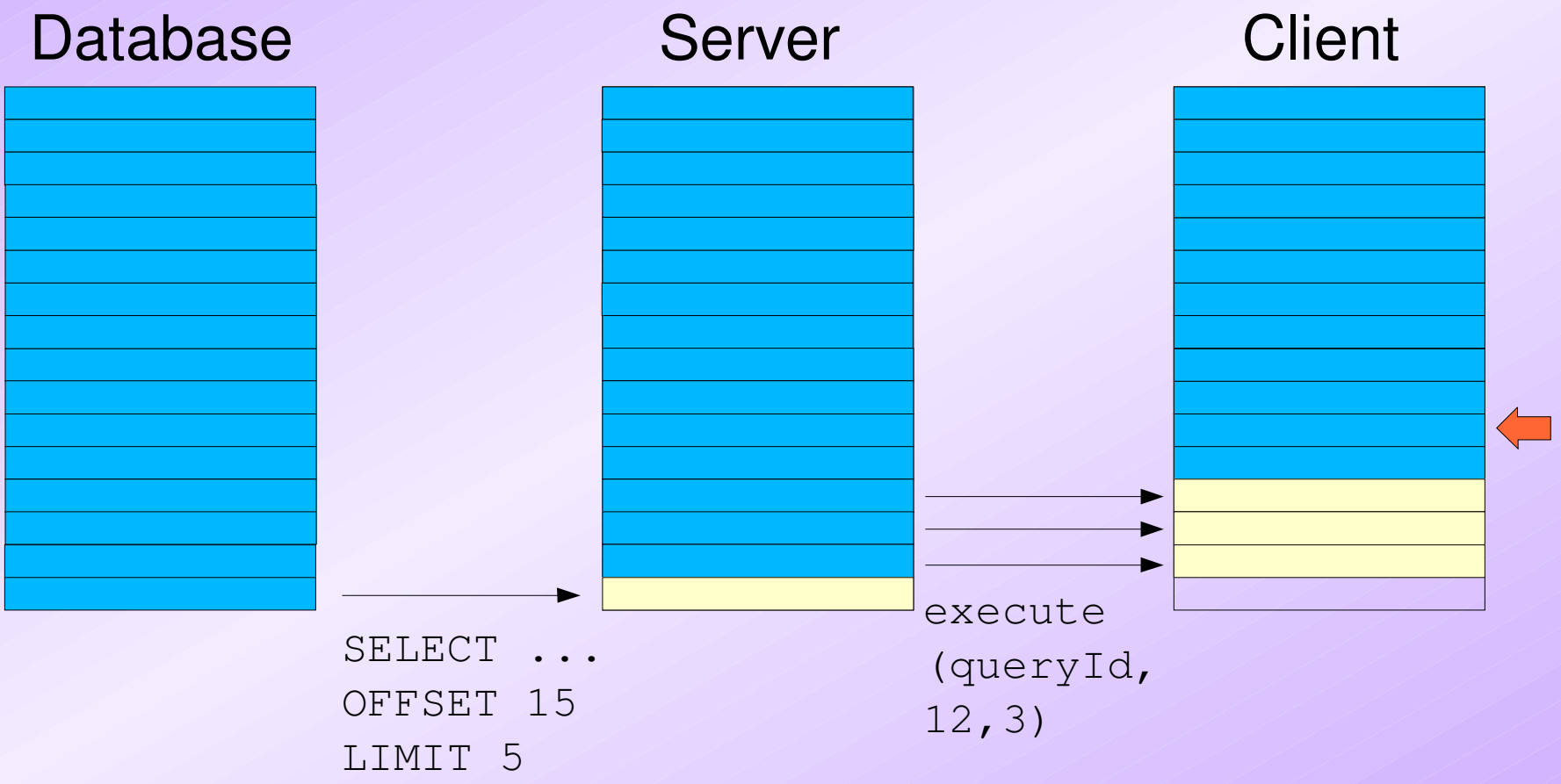
Server



Client



# Results


Database
Server
Client
SELECT ...
OFFSET 15
LIMIT 5
execute
(queryId,
12,3)

# Arbitrary queries – problems

- Badly formed queries may overload database server
- Difficult to optimise database for all queries
  - Which indexes to use?
  - Slow response to complex queries involving multi-table joins

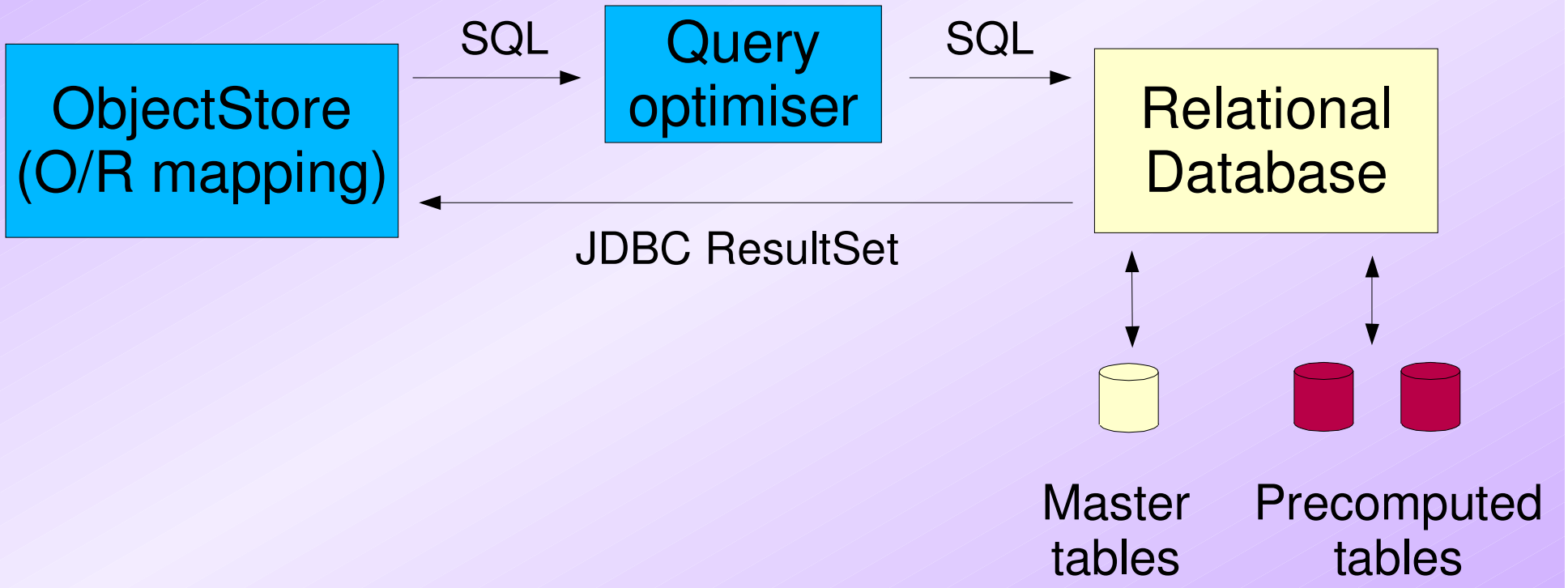
# Arbitrary queries – solutions

- Close relationship with database server query planner
  - ask how long a query will take before attempting to run it (~3ms)
  - Disallow queries that will take longer than a certain threshold
- Store data massively redundantly in “precomputed tables” and rewrite incoming queries on-the-fly.

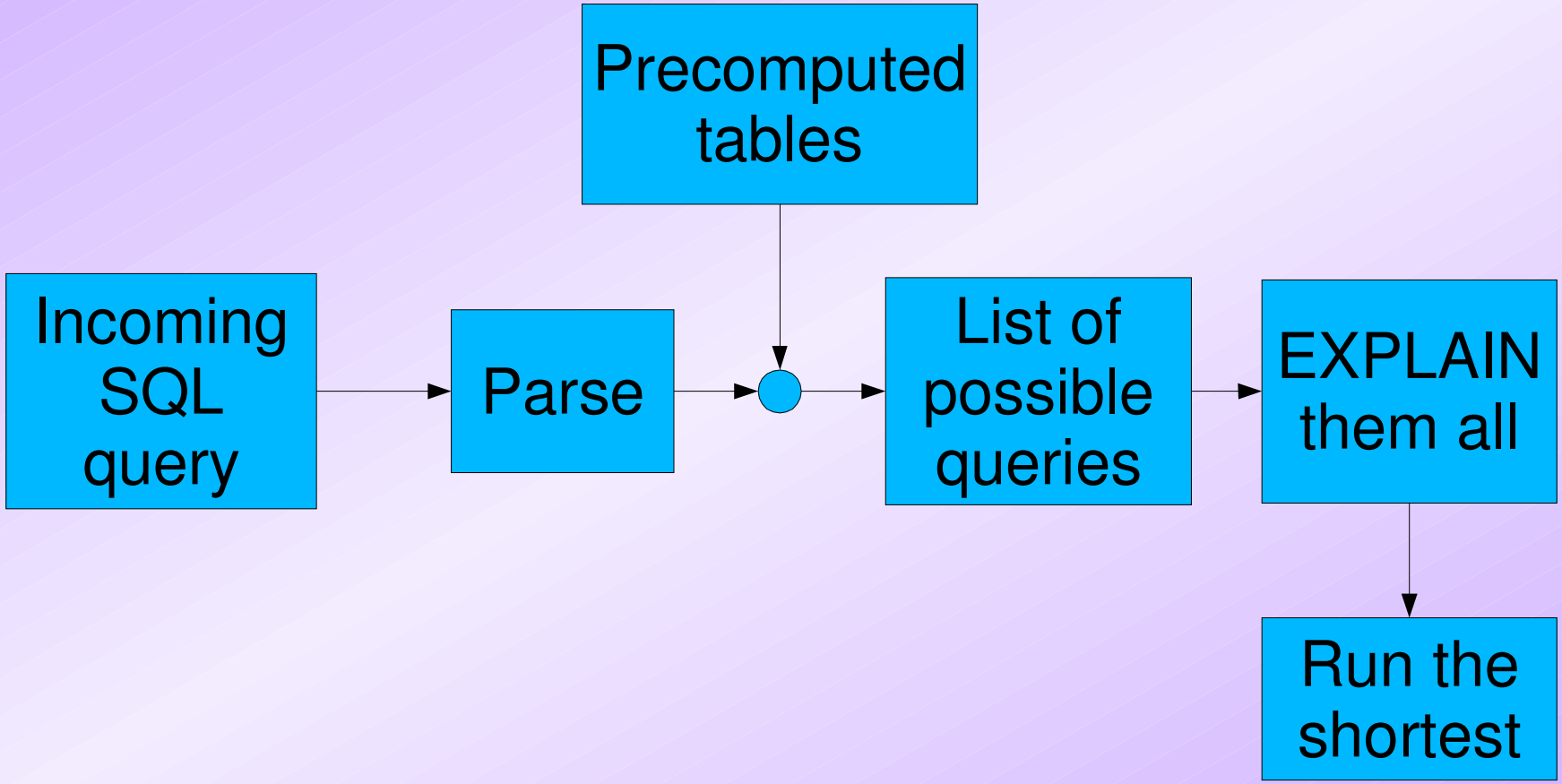
# Query optimisation – aims

- Provide a generic way of speeding up complex queries to any read-only SQL database
- Minimal parsing/computational overhead
- Transparent to users/applications
  - no new schema/model to learn
- Make available as standalone module

# Query optimisation architecture




# Query optimisation




# Query optimisation – example

## Master tables

Genes

| id | name |
|---|---|
| | |

GOTerms

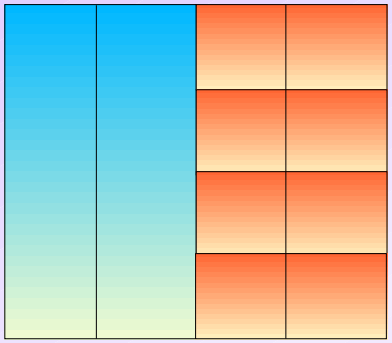
| geneid | code |
|---|---|
| | |

Experiments

| geneid | date | type |
|---|---|---|
| | | |

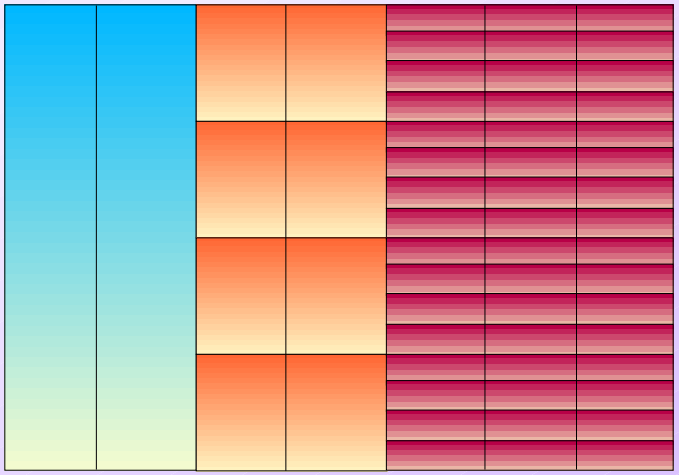
## Precomputed tables

GenesGOTerms



```
CREATE TABLE genesgoterms AS
SELECT genes.id AS genes_id,
       genes.name AS genes_name,
       goterms.geneid AS goterms_geneid,
       goterms.code AS goterms_code
FROM genes, goterms
WHERE genes.id = goterms.geneid
```

GenesGOTermsExperiments



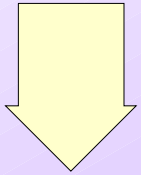
```
CREATE TABLE ....
SELECT ....
FROM genes, goterms, experiments
WHERE genes.id = goterms.geneid
AND genes.id = experiments.geneid
```

FlyMine

# Query optimisation – example

*"Show dates that gene expression experiments were performed on genes which have GO term GO:0000278 applied"*

```
SELECT genes.name, experiments.date
FROM genes, goterms, experiments
WHERE genes.id = goterms.geneid
AND genes.id = experiments.geneid
AND goterms.code = "GO:0000278"
AND experiments.type = "Gene Expression"
```

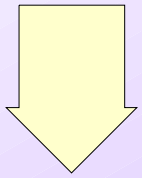


EXPLAIN

5 seconds

# Query optimisation – example

```
SELECT genes_name AS name,
       experiments.date
FROM genesgoterms, experiments
WHERE genes_id = experiments.geneid
AND goterms_code = "GO:0000278"
AND experiments.type = "Gene Expression"
```

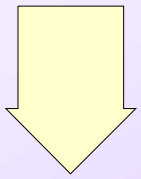


EXPLAIN

1 second

# Query optimisation – example

```
SELECT genes_name AS name,
       experiments_date AS date
FROM genesgotermsexperiments
WHERE goterms_code = "GO:0000278"
AND experiments_type = "Gene Expression"
```



EXPLAIN

200ms

# Query optimisation – example

- Without optimiser

  Time = 3ms + 5 seconds

  (EXPLAIN) (EXECUTE)

- With optimiser

  Time = 9ms + 100ms + 200ms

  (3 EXPLAINs) (PARSE) (EXECUTE)

# Query optimisation – summary

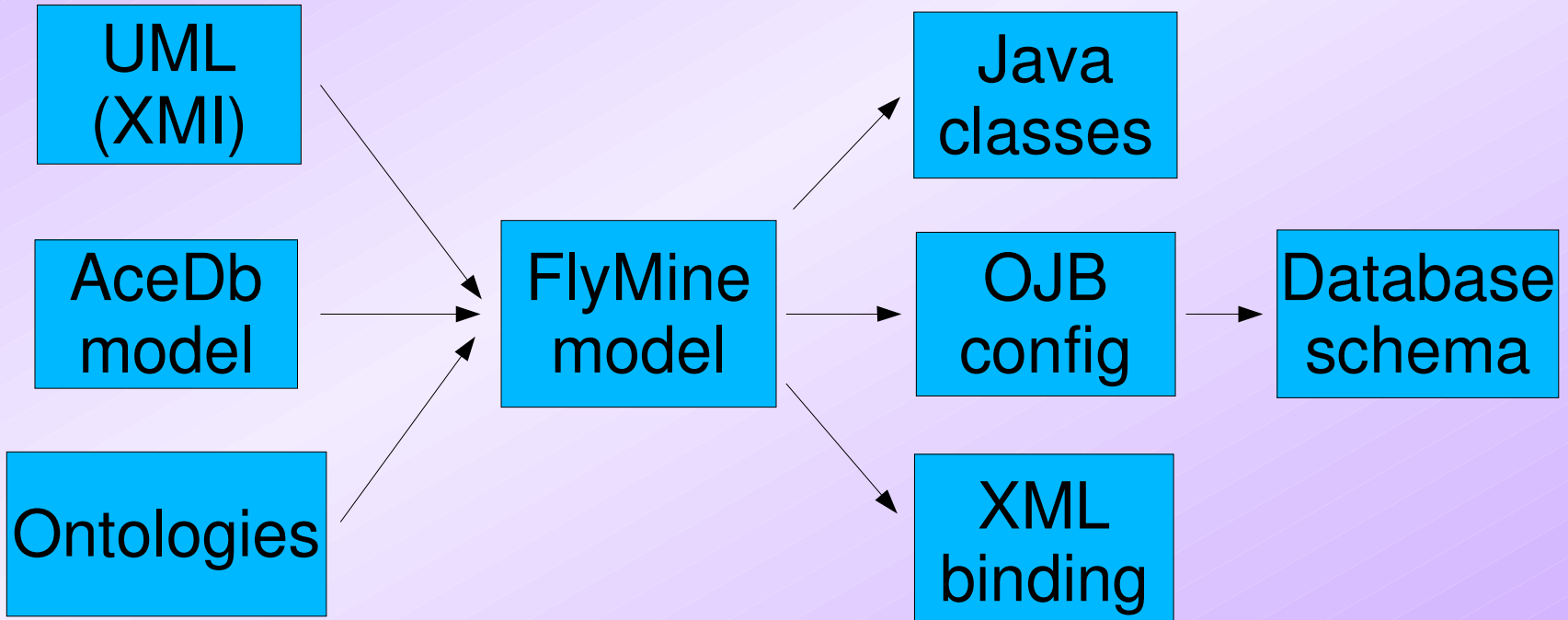
- For complex queries, the SQL optimisation module can produce large speed increases
- Optimisation is transparent to the user or application generating the SQL.
- Choosing which precomputed tables to store is important – may need to analyse incoming queries.

# Query optimiser – summary

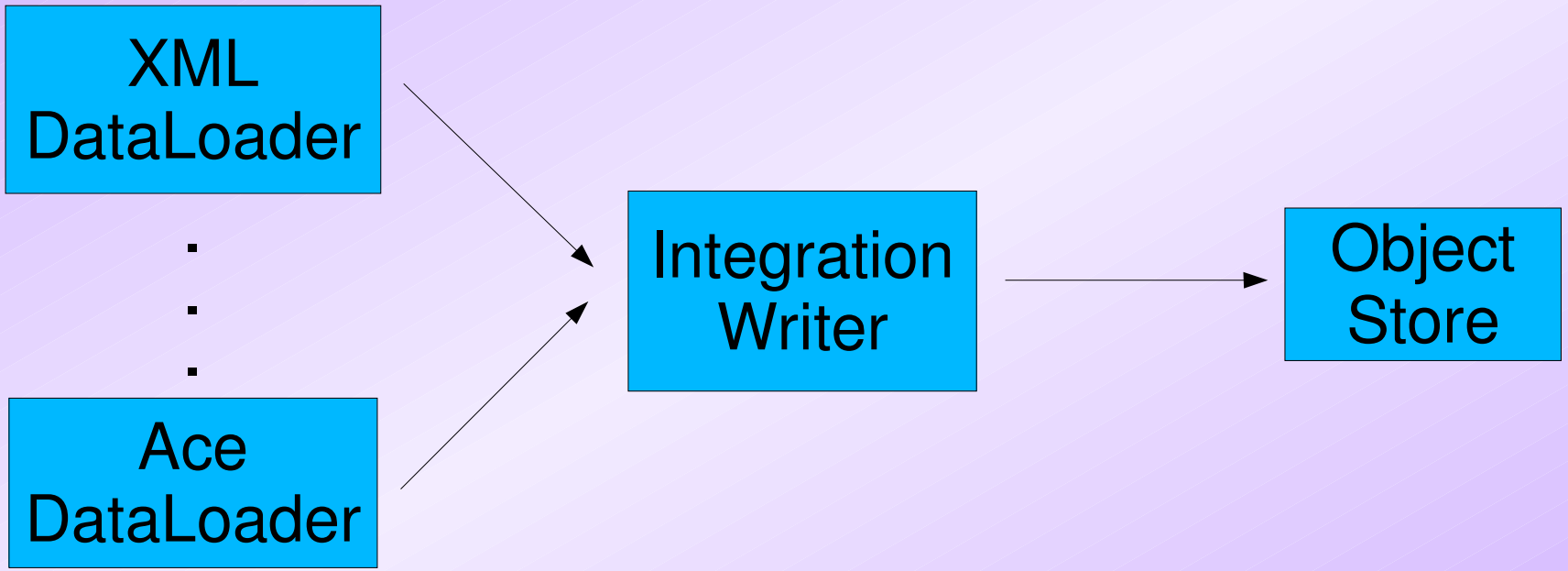
- Optimiser trades off disk space for improved query performance.
- Schema independent.
- Can be used in conjunction with P6Spy to intercept JDBC calls from existing software to a database.
- Available from www.flymine.org !!

# Data model

- Data model can be generated from one of a number of sources




# Data loading and integration




# Summary

- Current status
  - generic SQL query optimiser
  - powerful object database
  - 2 query interfaces (OQL + Java)
  - web service
  - framework for data loading/integration
  - autogeneration of model-dependent parts from UML or Ace model

# Summary

- Coming up
  - graphical query interface
  - web front end
  - integrated Drosophila/Anopheles database (genomics, microarray, proteomics, etc)
  - tools for auto-generation of best set of precomputed tables

# Acknowledgements

The FlyMine team:

Andrew Varley
Richard Smith
Matthew Wakeling
Mark Woodbridge
François Guillier
Rachel Lyne
Rajasekhar Paidi
Gos Micklem

Download FlyMine from www.flymine.org


FlyMine is funded by the Wellcome Trust (grant no. 067205), awarded to M. Ashburner, G. Micklem, S. Russell, K. Lilley and K. Mizuguchi.