

Common Interface for Every Kind of Astronomical Data Service

Yuji Shirasaki

National Astronomical Observatory
of Japan, JVO



JVO

VOQL+DAL Joint session

Contents

- Objective of this talk
- Concept for adapting ADQL query interface to a General data service
- JVO SkyNode experimental implementation
 - SkyNode Toolkit
 - Application to the science use case
- Required Specification
 - Data Service Classification and Table Data model
 - Definition of minimum subset of ADQL
 - Minor update on ADQL
 - Standard on usage of VOTable
 - Access to the Object data type

Objective of this talk

- From a Simple Protocol to an Intelligent Protocol
 - SIAP and SSAP are simple parameter-based query protocol.
 - ADQL is a SQL-based query protocol and have a capability to describe more complicated query conditions.
- What is the merit ?
 - Multiple regions condition, directly or by VOTable
 - Homogeneous data access to both of the catalog and observational data
 - Query to the multiple data services (Portal)
- Effect on the current SIAP and SSAP spec.
 - Introducing the ADQL to the DAL service does not affect to the data model of a returned VOTable.
 - Just introduce an additional query interface to the DAL service.

Concept for adapting SQL to Image Database.

I have shown that image query can be described in SQL syntax by introducing virtual column concept.

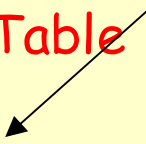
SIAP Query

POS [mandatory]
SIZE [mandatory]
FORMAT [mandatory]
INTERSECT [option]
...

```
http://jvo.nao.ac.jp/Image?  
Pos=34.3,-5.11&Size=0.01&Format=VOTable&  
INTERSECT=OVERLAPS
```

SIAP parameters and returned metadata are taken as columns of the virtual table.

Image Query on Virtual Table



Pos	Size	Format	ImageURL
(23,+30)	1.0	fits	http://jvo.nao.ac.jp/ Image?id=124214
(23,+23)	0.3	jpeg	http://jvo.nao.ac.jp/ Image?id=124215
...

```
Select imageURL  
From imageTable  
Where Pos = Point(23,+30)  
and Size = 1.0 and  
Format = 'fits'
```

This is a virtual table which has infinite number records.

Development of the JVO SkyNode Toolkit

- **Primary aim:**
 - to provide a reference implementation for every kind of data service using ADQL & VOTable interface
- **Supported DBMS:**
 - aimed to be independent on the type of DBMS, but still PostgreSQL native SQL is used...
 - The only requirement is availability of JDBC driver.
- **Restrictions:**
 - Not all the ADQL syntax are supported.
 - String representation of ADQL is JVOQL.
- **Experimental Release:**
 - <http://jvo.nao.ac.jp/download/skynode-toolkit/>

Software used

- Tomcat 4.1.31 ---- servlet container
- Axis 1.2RC1 ---- web service engine
- J2SDK 1.4.2 ---- Java compiler & library
- Ant 1.6.1 ---- Java-based build tool
- JavaCC 3.2 ---- parser generator for Java
- JAXB v1.0.3-b18-fcs ---- XML \leftrightarrow Java conversion
- PostgreSQL 7.4.7 ---- DBMS
- Java HTM library (JHU) ---- spherical indexing
- Java FITS library (HEASARC) ---- FITS IO lib
- ...

Architecture

JVO SkyNode Toolkit Flow Chart

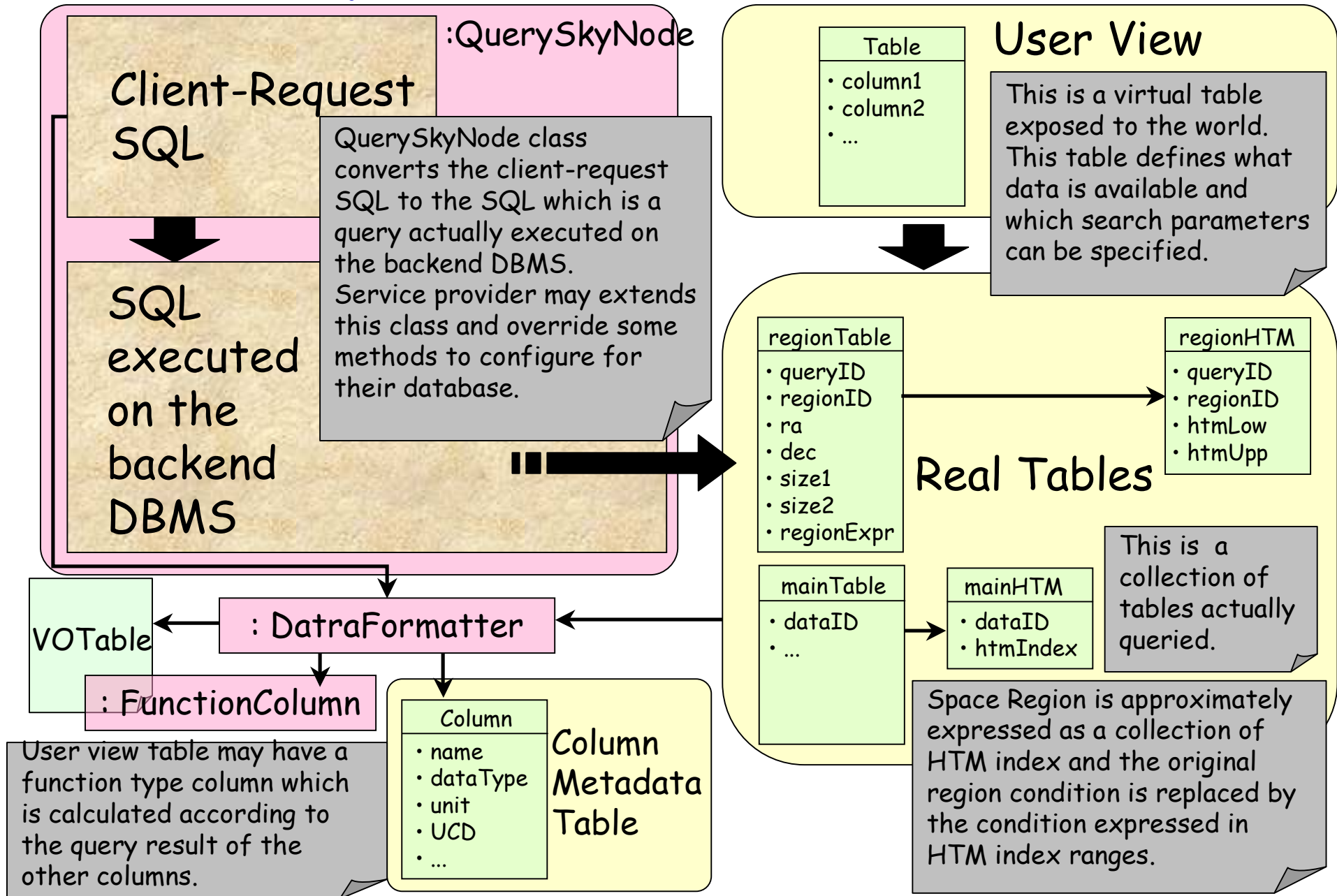


Image Data Query by ADQL

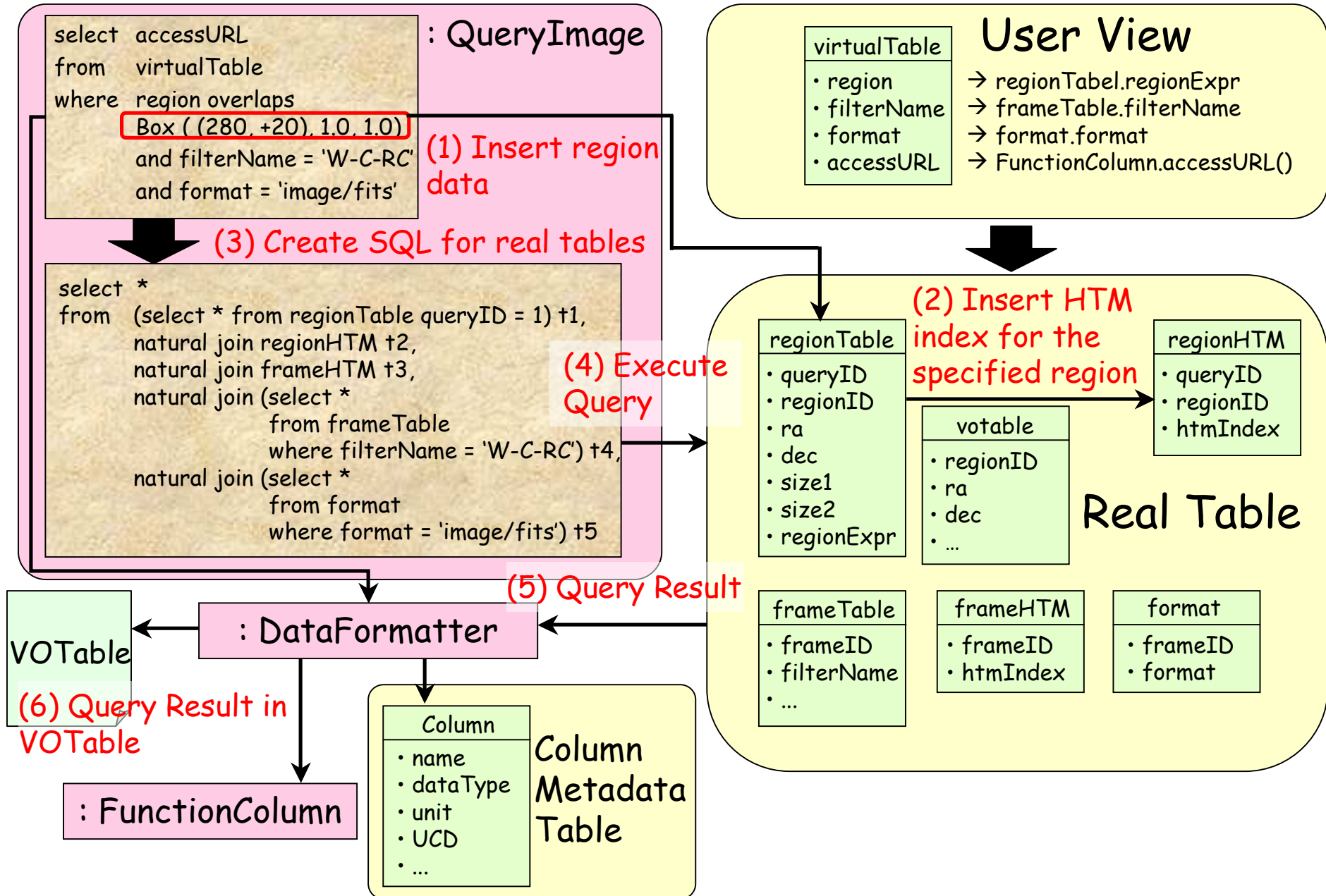
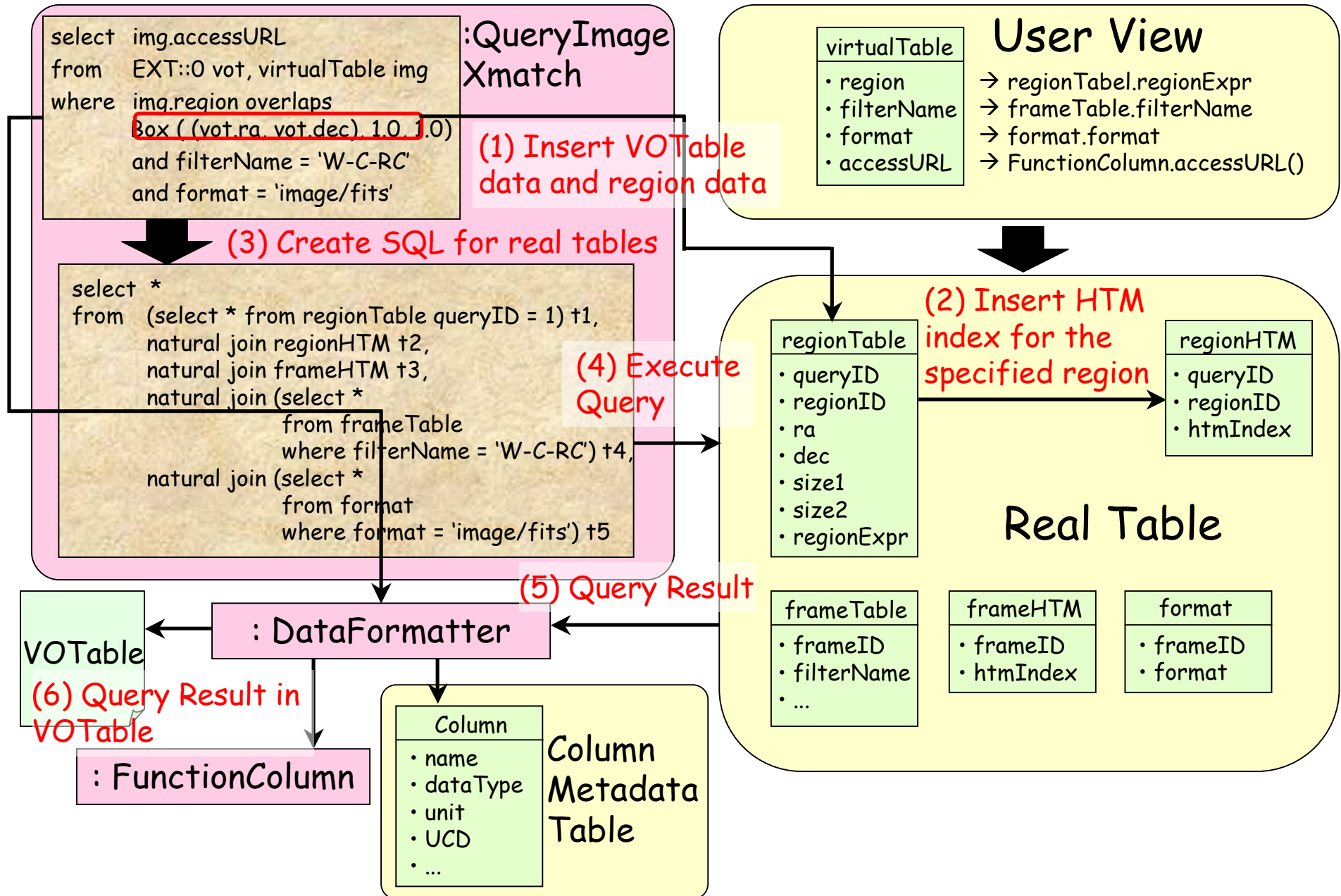


Image Data Xmatch Query with VOTable



Science Use Case
QSO/Galaxy
Clustering Study

Work flow for studying the QSO/Galaxy clustering

1. Select QSO coordinates from the QSO catalog → Query to the Skynode Catalog Database.
2. Search deep imaging data which covers the QSO regions → Query to the Skynode of Subaru Image Database
3. Create catalog from the imaging data → Invoke the SExtractor Web service.
4. Estimate the distance to the objects around the QSO → Invoke the HyperZ Web service
5. Try Clustering Analysis → Invoke the clustering analysis web service.

Query Flow Chart

1. User request SQL
2. SQL to the QSO catalog DB
3. QSO coordinates in VOTable
4. SQL & VOTable to Image DB
5. AccessURL for images around QSOs

```

SELECT cat.*, img.*
FROM jvo.misc.qso_veron_2003 cat,
     jvo.test.smoka.spcam_img img
WHERE POINT(cat.raj2000, cat.dej2000)
      WITHIN
      CIRCLE((189.20625, 62.216111), 0.10)
      AND cat.v_mag < 20
      AND img.format = 'image/fits'
      AND img.region OVERLAPS
      BOX((cat._raj2000, cat._dej2000), 0.02, 0.02)
    
```

(1)

```

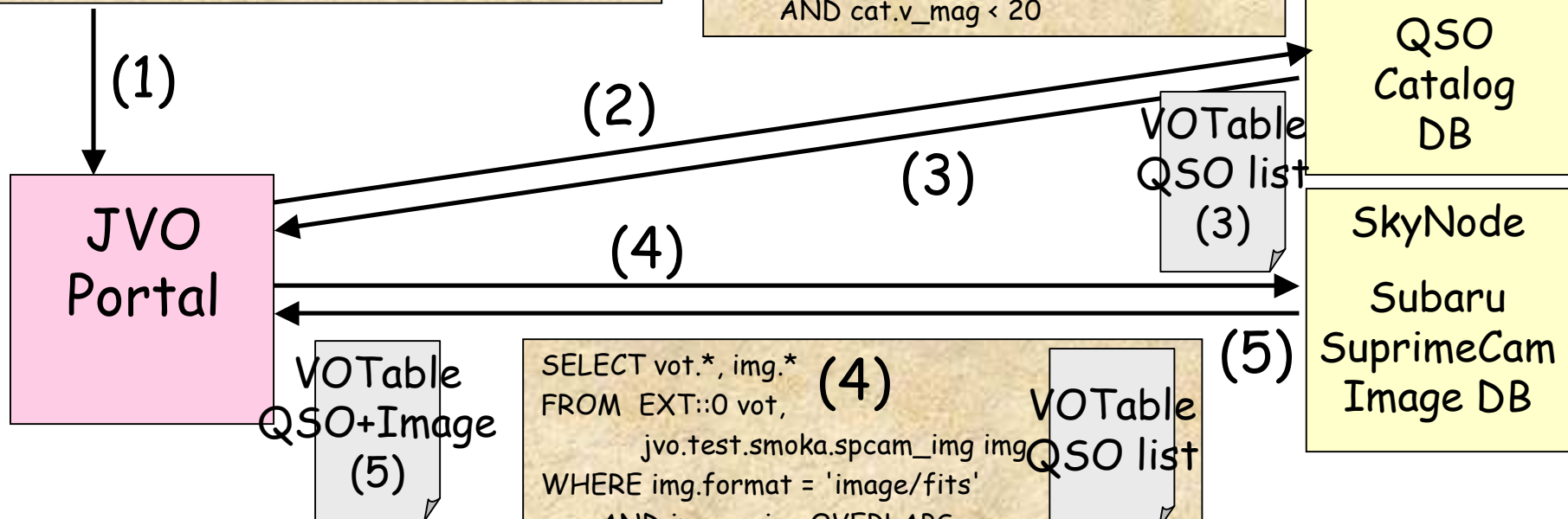
SELECT cat.*
FROM jvo.misc.qso_veron_2003 cat
WHERE POINT(cat.raj2000, cat.dej2000)
      WITHIN
      CIRCLE((189.20625, 62.216111), 0.10)
      AND cat.v_mag < 20
    
```

(2)

```

SELECT vot.*, img.*
FROM EXT:::0 vot,
     jvo.test.smoka.spcam_img img
WHERE img.format = 'image/fits'
      AND img.region OVERLAPS
      BOX((vot.raj2000, vot.dej2000), 0.02, 0.02)
    
```

(4)



QSO-Galaxies Search

[Status](#) | [Registry](#) | [Search](#) | [Result](#) | [Database](#) | [QSO Search](#) | [Image Viewer](#) | [Logout](#)
 ⇒ [Query](#) | [Catalog](#) | [PhotoZ](#)

Data Search

Search

ID for your query

Observation Name

Region

RA: [deg]
 Dec: [deg]
 Radius: [deg]
 Image Size: [arcmin]

Brightness

V_mag between [mag] and [mag]

Redshift

z between [mag] and [mag]

QSO Catalog Search

Service For QSO
 Table For QSO

Image Data Service Search

Service For Image
 Table For Image

Search

User ID	User Name	Group	Last Login
yshirasa	Yuji Shirasaki	jvo	Sat Mar 12 21:36:49 JST 2005

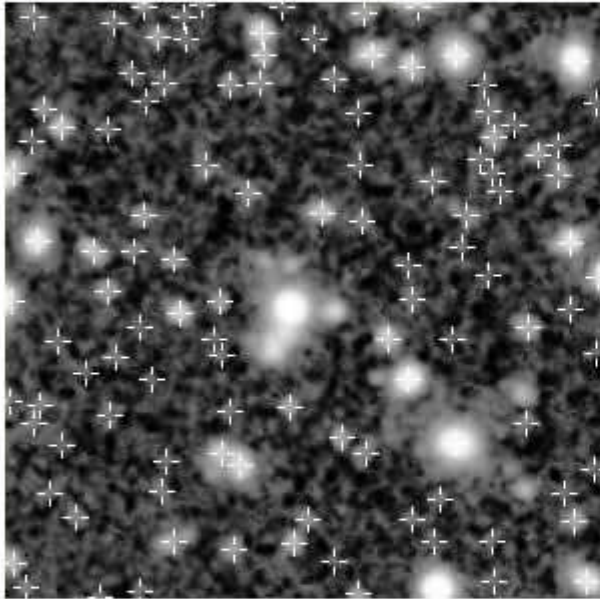
Image Viewer - Mozilla Firefox

ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)

Image Viewer

[Status](#) | [Registry](#) | [Search](#) | [Result](#) | [Database](#) | [QSO Search](#) | [Image Viewer](#) | [Logout](#)

Name	Origin	Scale	Contrast
fits0	http://erida.dc.nao..	hist	min = 0.0 max = 65535.0 auto = true



Scale :

Contrast :

min =

max =

VOTable :

User ID	User Name	Group	Last Login
yshirasa	Yuji Shirasaki	jvo	Tue Mar 29 22:16:43 JST 2005

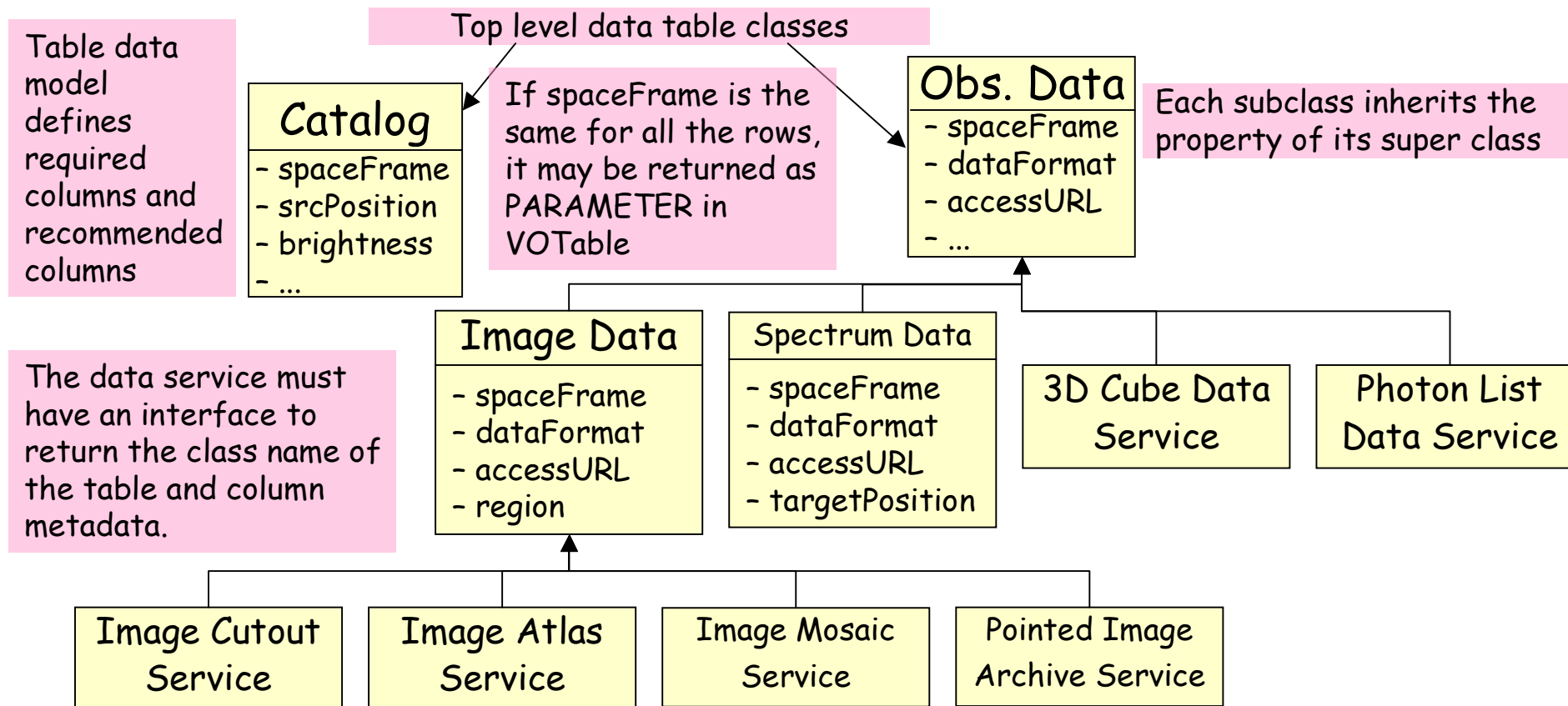
Total memory = 266403kB Used momory = 161121kB (60%)

JVO
JAPANESE VIRTUAL OBSERVATORY

Required Standard

Table Data Model for Astro-Data Service

- Query is described in SQL construct (ADQL).
- Query result is returned in a tabular form (VOTable).
- To write a SQL, table structure (column name) need to be known in advance.
- Table Data Model for each data service are required.



Column metadata

- All the column must have the following attributes:
 - **name, datatype, unit, arraysize, width, precision, ucd, utype**. These are used in the FIELD element attributes of VOTable. A coordinate column must have an **coordinateFrame** attribute.
 - column name can be specified by a data provider.
 - datatype must follows the VOTable specification.
 - expression of unit must follows the VOTable spec.
 - utype is supplied by VO data model. For columns not defined in the data model, utype is not required.
 - appropriate ucd is assigned by the data provider.

Minimum Requirement for Query Spec.

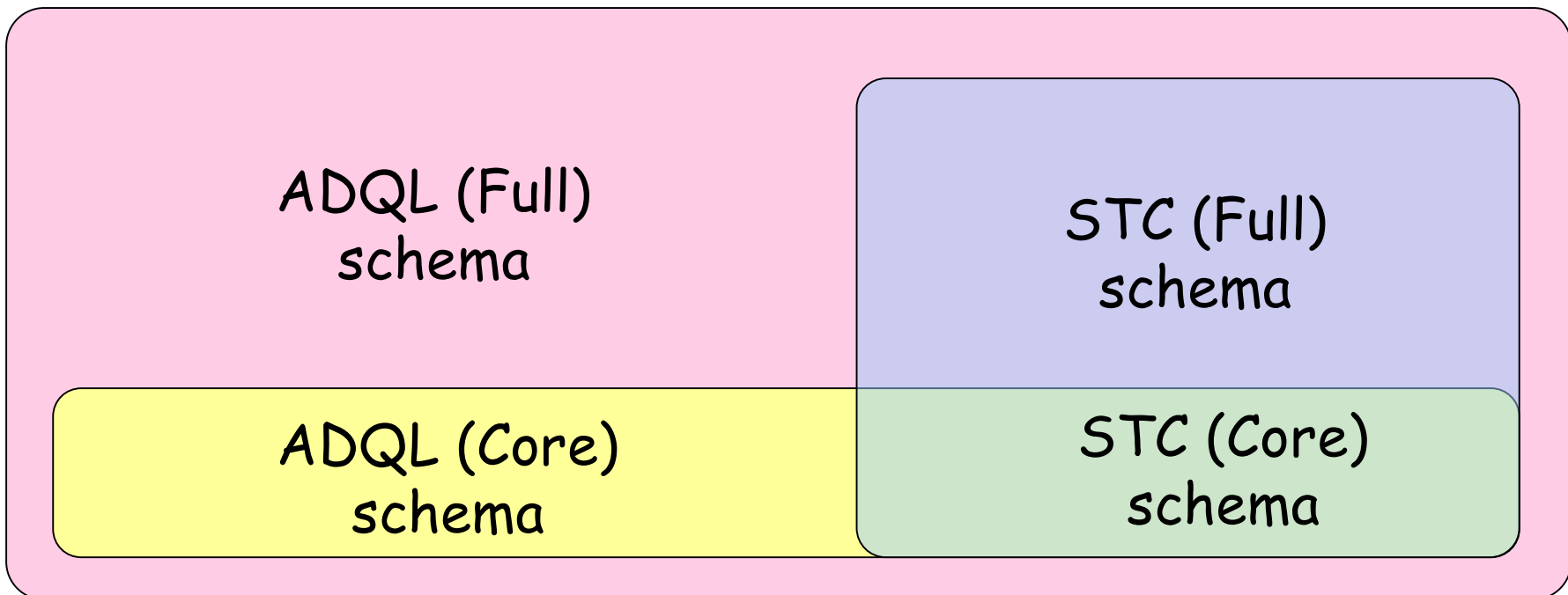
Current ADQL:

- 😊 describe most of the SQL92 syntax and some astro-extension
- 😞 many elements (33) & many data types (69), hard to build a data service. It is helpful to define minimum subset ADQL spec as a base line

- **SelectionItemType** (13 sub types in ADQL 0.8)
 - Only **AliasSelectionItem** and **ColumnReferenceType**
- **FromTableType** (3 sub types in ADQL 0.8)
 - Only **TableType**
- **SearchType** (16 sub types in ADQL 0.8)
 - Only **IntersectionSearchType** and **ComparisonPredType**
 - Region Search Criterion can be specified by **ComparisonPredType**.

Proposed ADQL Schema structure

- STC (Full) schema imports STC (Core) schema
- ADQL (Core) imports STC (Core)
- ADQL (Full) imports ADQL (Core) and STC (Full)
- ADQL (Core) and STC (Core) should not be changed. These are critical standard for interoperability.
- Each schema has a difference namespace.



Substitution of Column Name by UCD or UType

- Without knowing the column names, we cannot write an SQL...
- Possible Solution → use UCD or Utype for describing the query condition and selection list.
- Introduce "ucd" and "utype" attributes to the `columnReferenceType`

```
<Item xsi:type="columnReferenceType" Name="ra" Table="qso"/>
```

```
<Item xsi:type="columnReferenceType" ucd="pos.eq;src" Table="qso"/>
```

```
<Item xsi:type="columnReferenceType" utype="Target.pos" Table="qso"/>
```

ComparisonPredType as a "must-support" ConditionType

- A basic SkyNode MUST support "comparisonPredType".
- A basic SkyNode MAY support the other searchType.
- A basic SkyNode MUST recognize an unsupported searchType as a "trueType".
- A basic SkyNode MUST support the following construct:
 - <STC_columnName> <STC_operator> <STC_searchLocationType>
 - STC_operator ::= within, overlaps, outside
 - point within STC('Circle ICRS 200 -20 2.0') (Catalog Query)
 - region overlaps STC('Circle ICRS 200 -20 2.0') (Image Query)
 - spectrum overlaps STC('SpectralInterval A 4000 7000') and observationTime within STC('TimeInterval 2004-05-01 2004-05-31') (Spectrum Query)

STC-type column must be compared with an object of STC searchLocationType

utype defines which column represents STC.

ADQL-x version of point within 'Circle ICRS 200 -20 2.0'

It might be convenient to define substitution types for frequently used frames such as `<SpaceFrame xsi:type="ICRSFrameType"/>`

```
<Condition comparison="within" xsi:type="comparisonPredType">  
  <Arg Table="t" utype="src.position" xsi:type="columnReferenceType"/>  
  <Arg xsi:type="searchLocationType">  
    <AstroCoordSystem ID="ICRS">  
      <SpaceFrame>  
        <ICRS/><BARYCENTER/><SPHERICAL coord_naxes="2"/>  
      </SpaceFrame>  
    </AstroCoordSystem>  
    <AstroCoordArea ID="SearchRegion" coord_system_id="ICRS">  
      <Region>  
        <Circle unit="deg"> <Center>200 -20</Center> <Radius>2.0</Radius> </Circle>  
      </Region>  
    </AstroCoordArea>  
  </Arg>  
</Condition>
```

Which frame should be defined as a "must be supported" frame

Returned VOTable

- **FIELD id** → If alias name is specified "adql:aliasName", otherwise "adql:tableAliasName.columnName".
- **TABLE id** → table alias name
- **FIELD & TABLE name** → free, client app may use it
- **STC metadata (CoordSys)** for STC column
 - If each record in the STC column has a different STC metadata, STC metadata column must be included. Otherwise STC metadata must be included as PARAMETER.
- Some of the columns must always be included in the output VOTable. These columns are defined in SIAP, SSAP and SXAP document.

How to treat object data

- Spectrum metadata data has a structured data type. Access method to an object data is defined in SQL99. Example is:

Select t.*

From spectrumTable t

Where t.target.position within STC(' ... ')

(*) target is a column name, position is a field name of Target object. Always specify alias table name to express a column.

- Most of the recent RDB support object data type.
- If you don't want to use an object data type of RDB, relate one object type to one table or just have a flat table, then emulate object datatype column.