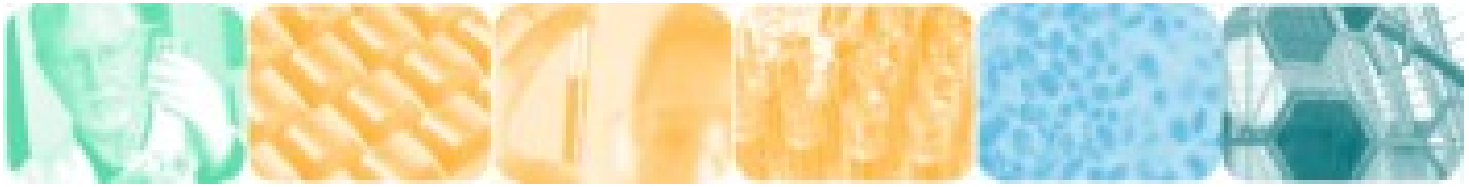
Instant JChem

Chemistry on your desktop

Tim Dudgeon <tdudgeon@chemaxon.com>



IJC team

IJC Core

- Petr Hamernik * - everything!
- Petr Zajac * - DB features
- Miloslav Sram – user features
- Martin Adamek – IJC server
- Max Sauer * - query
- Daniel Butler * - docs + QA
- Martin Krauskopf * - just started!

Also

- James Illston
- Tim Miller *

Marvin/JChem

- Csizi *
- Szilard Dorant *
- Gyorgy Pirok *
- Peter Kovacs *
- Szabolcs Csepregi *
- Tamas Vertse *
- Istvan Cseh *
- … and many many many many many many many many more

Topics

- What is IJC
- What is new since last year
- IJC databases - case studies

What is IJC

- Easy to use chemical workbench
- Aimed at chemists and biologists, not computational chemists
- Chemical search and reporting
- Personal and shared databases
- Collaboration and security

New since last year

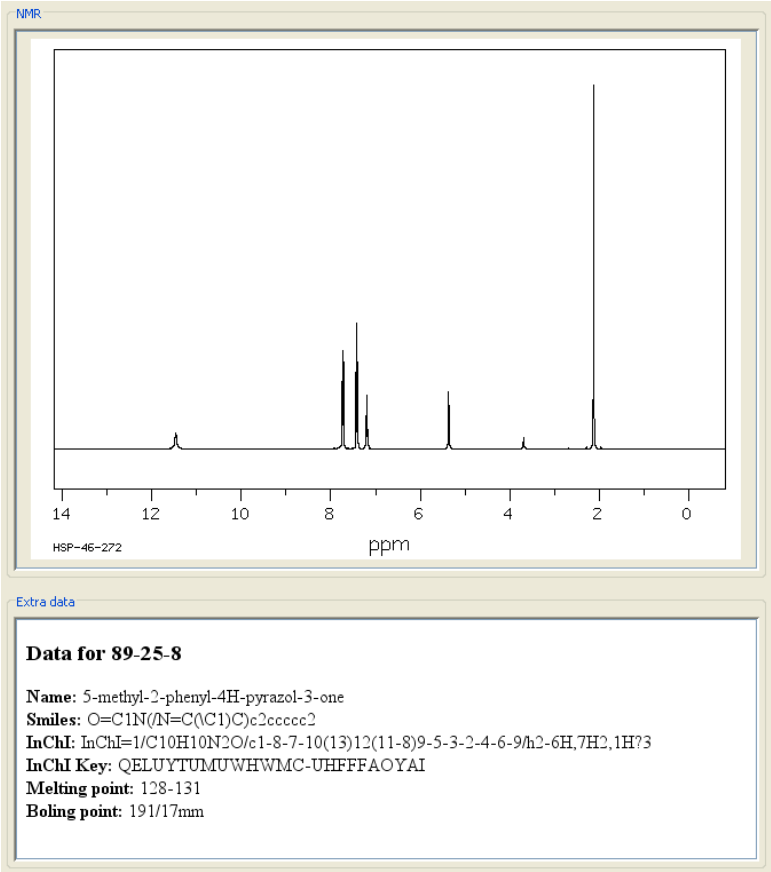
Several important features added since last year's UGM

Improved database features

- Support for multiple Oracle schemas
- Support for text based primary keys
- Support for database views
- Improved support for JChem cartridge
- Improved schema editor
- Additional options for security

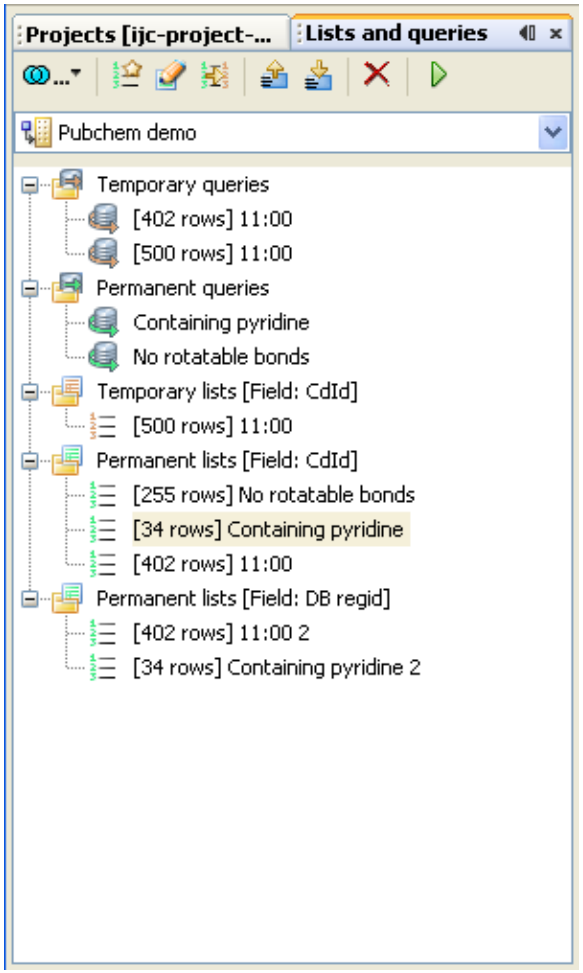
URL fields

- Dynamic or statically generated URL fields
- Pull in data from internet
- Link out to internet
- Browser widget allows display of HTML or images
- Content can be distributed within the IJC project



List management improvements

- Lists for any integer or text field
- Convert lists between different fields
- Copy/paste
- Drag and drop



Miscellaneous

- Improved printing
- Reactor integration
- Additional similarity search metrics
- Improved look and feel
- More flexible structure renderer
- Import of Markush DARC files

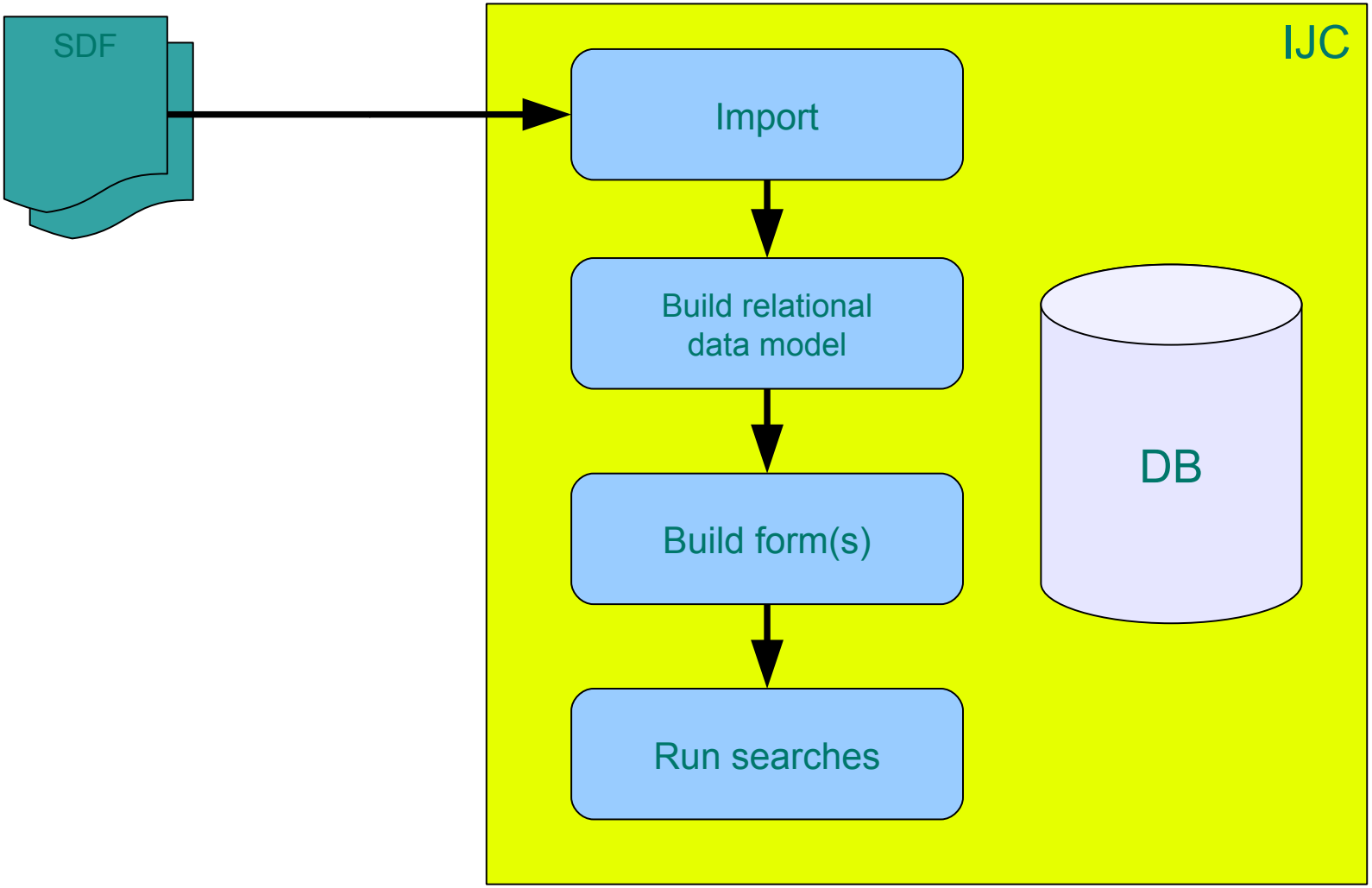
Future features

- More database features
- … and more still
- IJC server
- Visualisation
- Calculations
- LogP/pKa/... training

Using IJC to deploy or access chemical databases

1. Reporting from Activity Base
2. Supplying DBs to partners/customers
3. Markush/Patent DBs

Standard usage



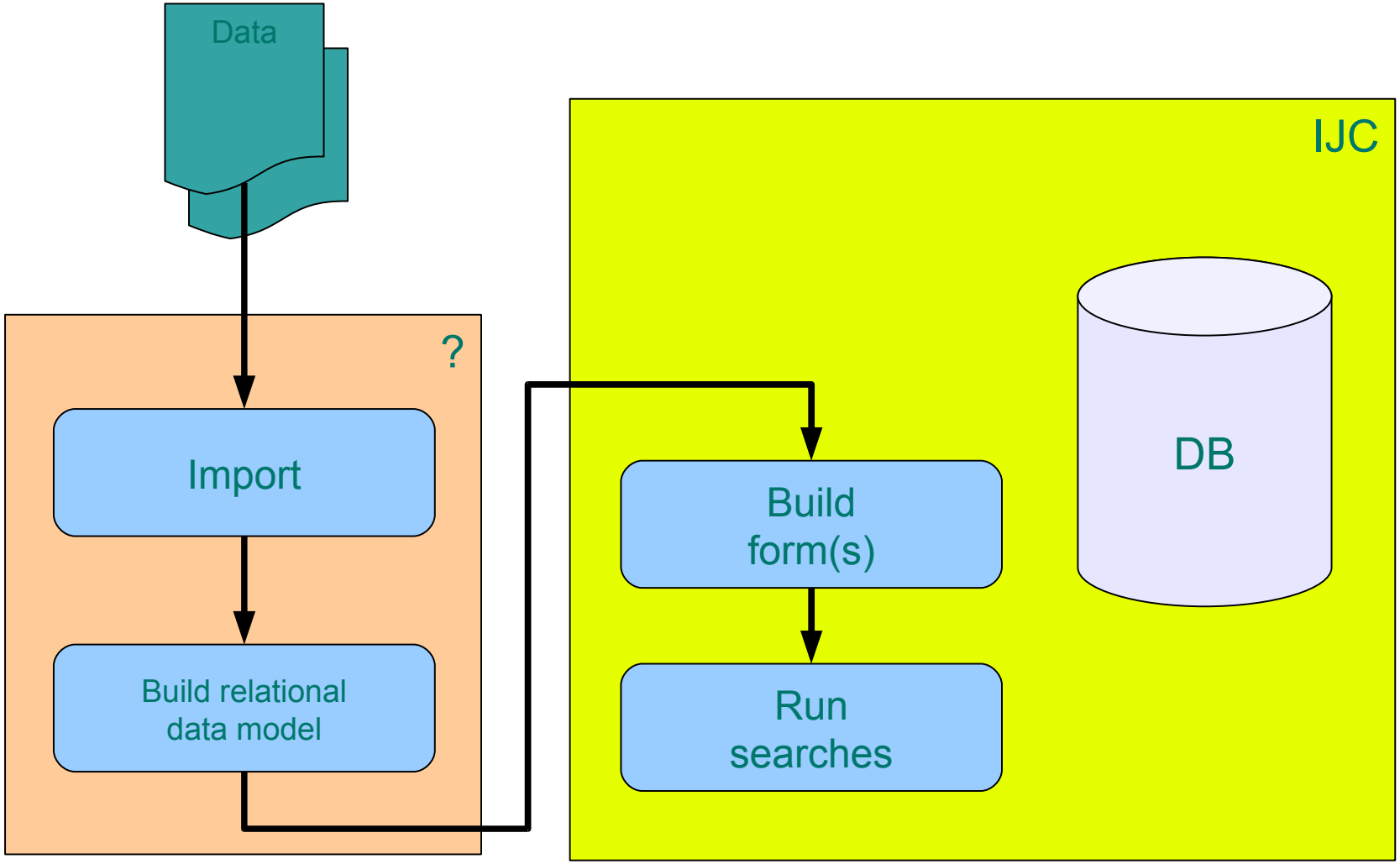
Limitations

- Import is not infinitely flexible
- Data model is not infinitely flexible
- Does not allow custom data processing
- Cannot be automated
- ....

  → We are working to remove these limitations

  → But what to do now when you run into the buffers?

Paradigm for non-standard usage



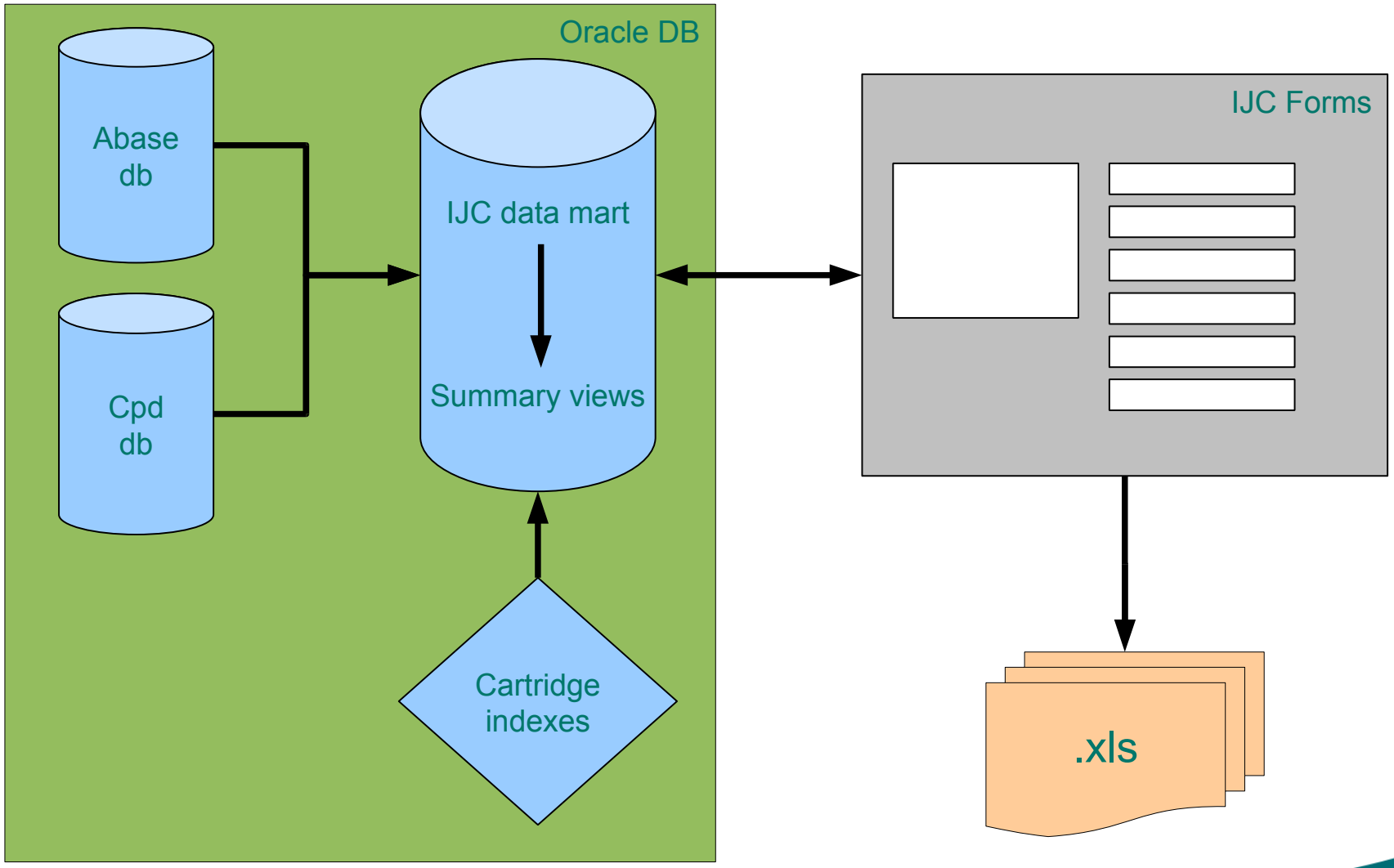
Data processing external to IJC

- Database is a standard SQL database
  - Nothing hidden, even for local databases
  - Can be manipulated by anything that can use JDBC and JChem API (and maybe even .NET ....)
- Allows much more flexibility and automation
- Many approaches possible
  - SQL
  - Pipeline Pilot, Knime...
  - Scripts/programs
- Once processing is complete data can be picked up and used by IJC

1. Reporting chemical and assay data

- Biotech company with assay data in Activity Base
- Lots of data, but no good way of viewing, searching, or distributing it
- Needed simple, easy to use, but inexpensive solution

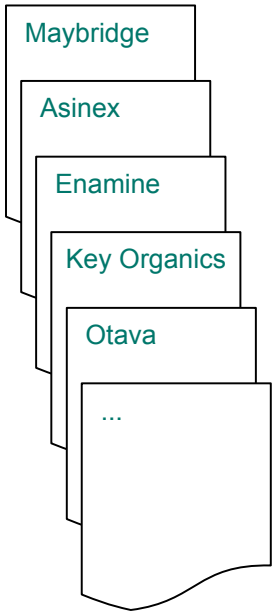
Solution


Oracle DB
Abase db
Cpd db
IJC data mart
Summary views
Cartridge indexes
IJC Forms
.xls

Use of materialised views

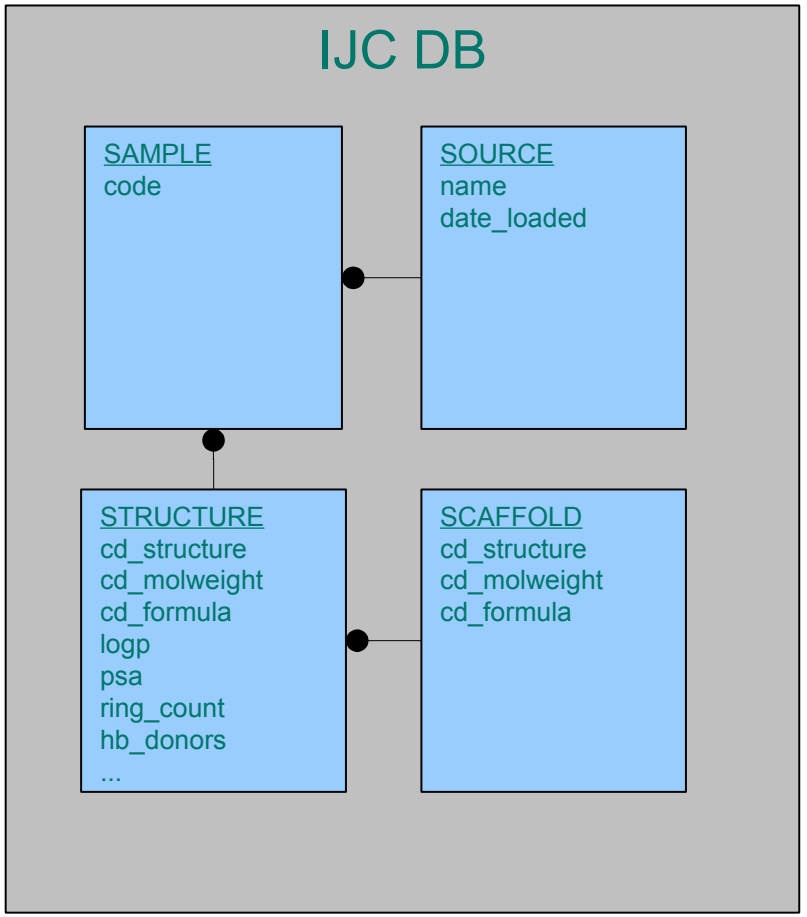
- Allows data from highly normalised data model to be converted to a form better suited to reporting
- Provides fast access to data
  - Data is already pivoted
  - Indexes added as needed
  - Summarized (averaged) data available
- JChem cartridge indexes can be added to materialised views (e.g. smiles column)
- Materialised views refreshed as needed (e.g. every night)

2. Suppliers database



Import

Duplicate checking
Standardization
Chemical terms columns
BMF scaffolds
...



Groovy loader scripts

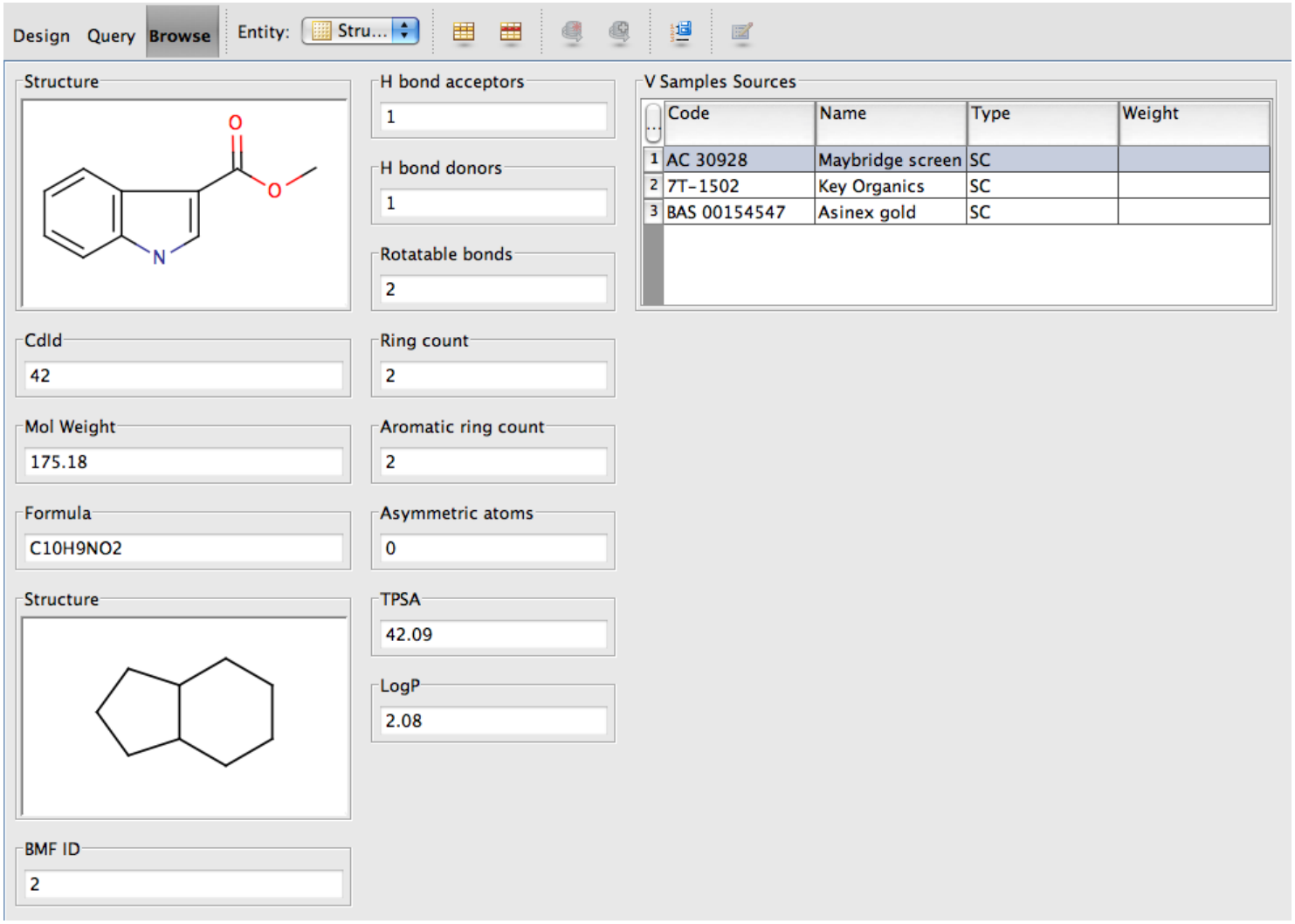
- Use groovy-jchem.jar library that wraps some Marvin/JChem functions and makes them easier to use
  - SD file parsing (MRecordReader)
  - JChem table handling (UpdateHandler etc.)
- Each SD file has simple configuration to allow it to be loaded by core loader class
- groovy.sql.Sql class makes loading using SQL a breeze

Groovy script

- Creates tables
- Parses SDF
- Calculates BMF
- Loads into 4 main tables (+ 1 view)
- < 200 lines code in core loader class

```
…

jchemproperties = new PropertyTable("JCHEMPROPERTIES", sql.connection)
def structures = new JChemBaseTable("STRUCTURES", jchemproperties)
def structuresBMF = new JChemBaseTable("STRUCTURES_BMF", jchemproperties)
def parser = new StructureParser(fileName)

...

parser.each() {
        sql.withTransaction() {
            try {
                def mol = parser.moleculeAsString

                def bmfId = newStructureToBeRegistered(mol)

                structuresUH.structure = mol
                structuresUH.setValueForAdditionalColumn(1, bmfId)
                def cd_id = structuresUH.execute(true)
                def id = cd_id
                if (id < 0) {
                    id = -id // duplicates are negative IDs
                    dups++
                } else {
                    nonDups++
                    //newStructureRegistered(id, mol)
                }

                def code = parser.getPropertyAsString(fieldName)
                sql.executeInsert(samples, [sourceId, id, code])
            } catch (Exception e) {
                println "Failed to load record $parser.counter"
                e.printStackTrace()
                failures++
            }
        }
        if (parser.counter % 1000 == 0) {
            println "$parser.counter records loaded..."
        }
}

...
```

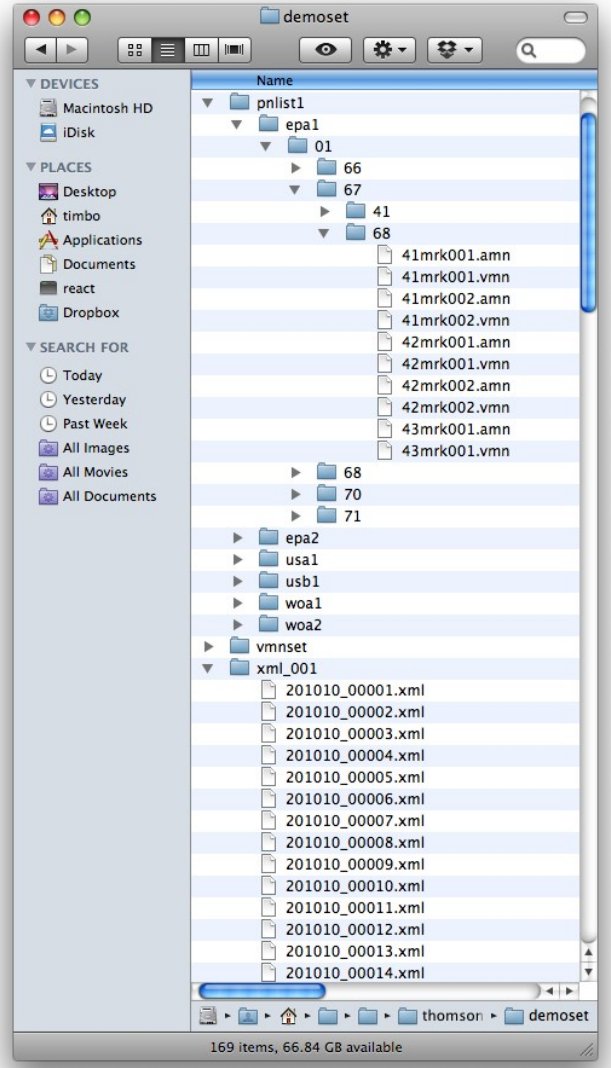
IJC database

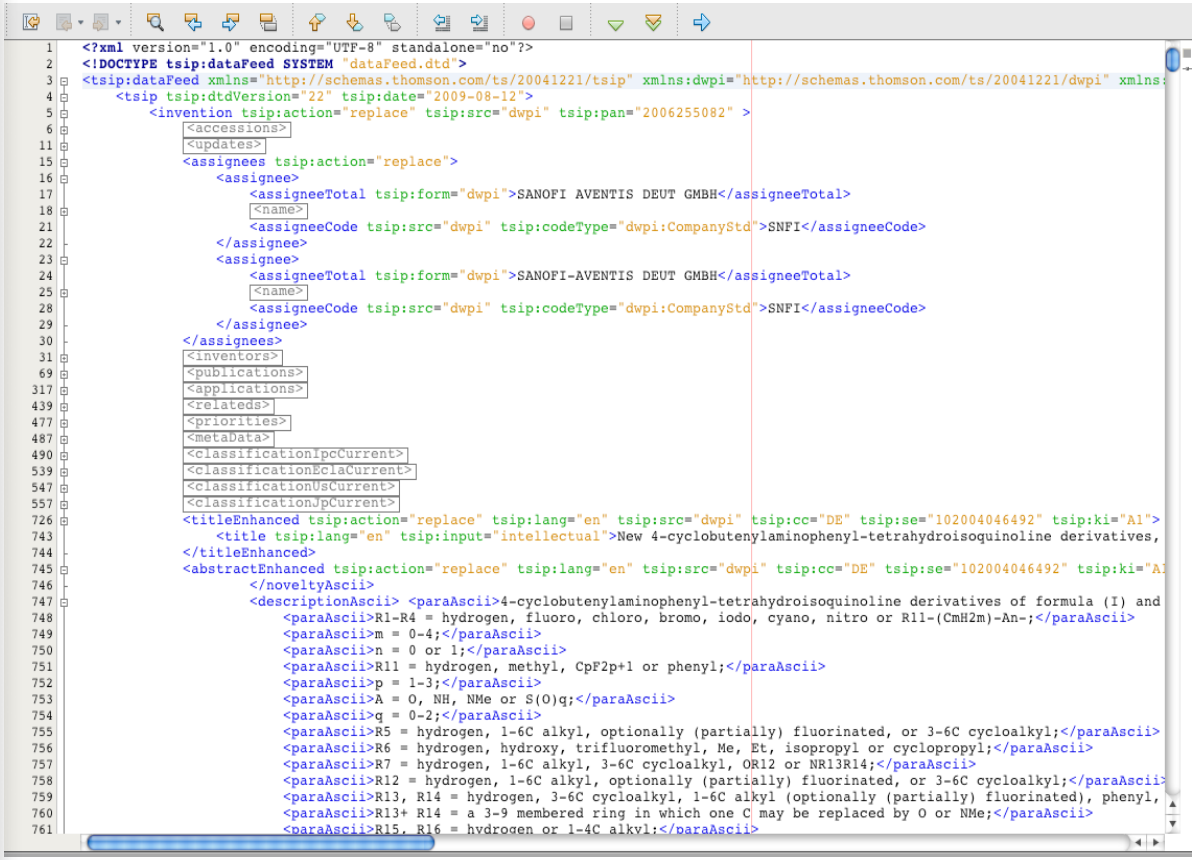


3. Markush demoset

- Markush demo data from Thomson Reuters
  - VMN files containing Markush structures
  - XML files containing patent data
  - Other files (images, PDFs...)
- IJC would be ideal reporting tool for this data
  - But can't import the data as its too complex
- Quick solution needed to allow customers to evaluate Markush technology + data

The data


demoset
DEVICES
Macintosh HD
iDisk
PLACES
Desktop
timbo
Applications
Documents
react
Dropbox
SEARCH FOR
Today
Yesterday
Past Week
All Images
All Movies
All Documents
Name
pnlist1
epa1
01
66
67
41
68
41mrk001.amn
41mrk001.vmn
41mrk002.amn
41mrk002.vmn
42mrk001.amn
42mrk001.vmn
42mrk002.amn
42mrk002.vmn
43mrk001.amn
43mrk001.vmn
68
70
71
epa2
usa1
usb1
woa1
woa2
vmnset
xml_001
201010_00001.xml
201010_00002.xml
201010_00003.xml
201010_00004.xml
201010_00005.xml
201010_00006.xml
201010_00007.xml
201010_00008.xml
201010_00009.xml
201010_00010.xml
201010_00011.xml
201010_00012.xml
201010_00013.xml
201010_00014.xml
thomson
demoset
169 items, 66.84 GB available


<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE tsip:dataFeed SYSTEM "dataFeed.dtd">
<tsip:dataFeed xmlns="http://schemas.thomson.com/ts/20041221/tsip" xmlns:dwpi="http://schemas.thomson.com/ts/20041221/dwpi" xmlns:
<tsip tsip:dtdVersion="22" tsip:date="2009-08-12">
<invention tsip:action="replace" tsip:src="dwpi" tsip:pan="2006255082" >
<accessions>
<updates>
<assignees tsip:action="replace">
<assignee>
<assigneeTotal tsip:form="dwpi">SANOFI AVENTIS DEUT GMBH</assigneeTotal>
<name>
<assigneeCode tsip:src="dwpi" tsip:codeType="dwpi:CompanyStd">SNFI</assigneeCode>
</assignee>
<assignee>
<assigneeTotal tsip:form="dwpi">SANOFI-AVENTIS DEUT GMBH</assigneeTotal>
<name>
<assigneeCode tsip:src="dwpi" tsip:codeType="dwpi:CompanyStd">SNFI</assigneeCode>
</assignee>
</assignees>
<inventors>
<publications>
<applications>
<relateds>
<priorities>
<metaData>
<classificationIpcCurrent>
<classificationEclaCurrent>
<classificationUsCurrent>
<classificationJpCurrent>
<titleEnhanced tsip:action="replace" tsip:lang="en" tsip:src="dwpi" tsip:cc="DE" tsip:se="102004046492" tsip:ki="A1">
<title tsip:lang="en" tsip:input="intellectual">New 4-cyclobutenylaminophenyl-tetrahydroisoquinoline derivatives,
</titleEnhanced>
<abstractEnhanced tsip:action="replace" tsip:lang="en" tsip:src="dwpi" tsip:cc="DE" tsip:se="102004046492" tsip:ki="A
</noveltyAscii>
<descriptionAscii> <paraAscii>4-cyclobutenylaminophenyl-tetrahydroisoquinoline derivatives of formula (I) and
<paraAscii>R1-R4 = hydrogen, fluoro, chloro, bromo, iodo, cyano, nitro or R11-(CmH2m)-An-;</paraAscii>
<paraAscii>m = 0-4;</paraAscii>
<paraAscii>n = 0 or 1;</paraAscii>
<paraAscii>R11 = hydrogen, methyl, CpF2p+1 or phenyl;</paraAscii>
<paraAscii>p = 1-3;</paraAscii>
<paraAscii>A = O, NH, NMe or S(O)q;</paraAscii>
<paraAscii>q = 0-2;</paraAscii>
<paraAscii>R5 = hydrogen, 1-6C alkyl, optionally (partially) fluorinated, or 3-6C cycloalkyl;</paraAscii>
<paraAscii>R6 = hydrogen, hydroxy, trifluoromethyl, Me, Et, isopropyl or cyclopropyl;</paraAscii>
<paraAscii>R7 = hydrogen, 1-6C alkyl, 3-6C cycloalkyl, OR12 or NR13R14;</paraAscii>
<paraAscii>R12 = hydrogen, 1-6C alkyl, optionally (partially) fluorinated, or 3-6C cycloalkyl;</paraAscii>
<paraAscii>R13, R14 = hydrogen, 3-6C cycloalkyl, 1-6C alkyl (optionally (partially) fluorinated), phenyl,
<paraAscii>R13+ R14 = a 3-9 membered ring in which one C may be replaced by O or NMe;</paraAscii>
<paraAscii>R15, R16 = hydrogen or 1-4C alkyl;</paraAscii>

Solution

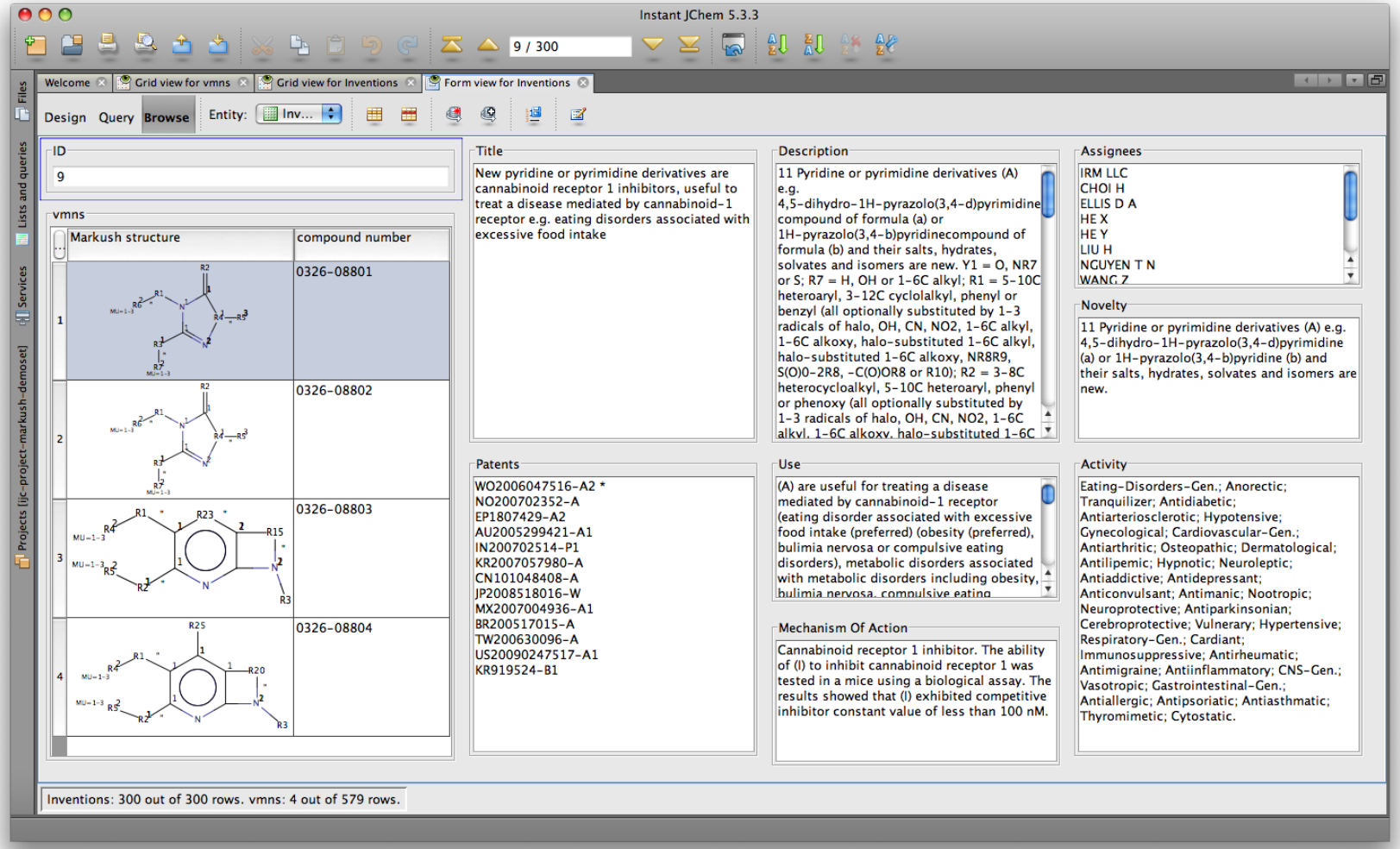
- Import VMN files using IJC file import
- Load data from the XML files into additional DB tables using custom groovy script
- Join the two sets together
- [Future] provide links to images and actual patent data
- Provide IJC project to customers as zip file

Groovy script

- Creates tables
- Trawls XML files
- Parses XML
- Assembles data
- Loads into DB
- < 200 lines code

```
...

File root = new File('/Users/timbo/data/structures/thomson/demoset/xml_001')
    def files = root.listFiles()
    def fileCount = 0
    def inventionCount = 0
    def markushCount = 0

    files.each {
        if (it.name.endsWith(".xml")) {
            println "Processing file $it"
            def dataFeed = new XmlParser().parse(it)
            fileCount++

            dataFeed.tsip.invention.each {
                inventionCount++
                //println "invention"
                String assignee = ""
                it.assignees.each {
                    it.assignee.assigneeTotal.each {
                        //println "  assignee = ${it.text()}"
                        assignee += it.text() + "\n"
                    }
                }

                def patents = ""
                it.publications.each {
                    it.publication.each {
                        def status = it.attribute(tsip.status)
                        def countryCode = it.documentId.countryCode.text()
                        def kindCode = it.documentId.kindCode.text()
                        def date = it.documentId.date.text()
                        def number = it.documentId.number.text()
                        def summary = "${countryCode}${number}-${kindCode}"
                        if (status == "dwpi:basic") {
                            summary += " *"
                        }
                        //println "  patent = ${summary}"
                        patents += "${summary}\n"
                    }
                }
                //println "  patents = \n${patents}"

...
```

Result


Instant JChem 5.3.3
9 / 300
Welcome
Grid view for vmns
Grid view for Inventions
Form view for Inventions
Design
Query
Browse
Entity:
Inv...
ID
9
vmns
Markush structure
compound number
0326-08801
0326-08802
0326-08803
0326-08804
Title
New pyridine or pyrimidine derivatives are cannabinoid receptor 1 inhibitors, useful to treat a disease mediated by cannabinoid-1 receptor e.g. eating disorders associated with excessive food intake
Description
11 Pyridine or pyrimidine derivatives (A) e.g. 4,5-dihydro-1H-pyrazolo(3,4-d)pyrimidine compound of formula (a) or 1H-pyrazolo(3,4-b)pyridinecompound of formula (b) and their salts, hydrates, solvates and isomers are new. Y1 = O, NR7 or S; R7 = H, OH or 1-6C alkyl; R1 = 5-10C heteroaryl, 3-12C cyclolalkyl, phenyl or benzyl (all optionally substituted by 1-3 radicals of halo, OH, CN, NO2, 1-6C alkyl, 1-6C alkoxy, halo-substituted 1-6C alkyl, halo-substituted 1-6C alkoxy, NR8R9, S(O)0-2R8, -C(O)OR8 or R10); R2 = 3-8C heterocycloalkyl, 5-10C heteroaryl, phenyl or phenoxy (all optionally substituted by 1-3 radicals of halo, OH, CN, NO2, 1-6C alkyl, 1-6C alkoxy, halo-substituted 1-6C
Assignees
IRM LLC
CHOI H
ELLIS D A
HE X
HE Y
LIU H
NGUYEN T N
WANG Z
Novelty
11 Pyridine or pyrimidine derivatives (A) e.g. 4,5-dihydro-1H-pyrazolo(3,4-d)pyrimidine (a) or 1H-pyrazolo(3,4-b)pyridine (b) and their salts, hydrates, solvates and isomers are new.
Patents
WO2006047516-A2 *
NO200702352-A
EP1807429-A2
AU2005299421-A1
IN200702514-P1
KR2007057980-A
CN101048408-A
JP2008518016-W
MX2007004936-A1
BR200517015-A
TW200630096-A
US20090247517-A1
KR919524-B1
Use
(A) are useful for treating a disease mediated by cannabinoid-1 receptor (eating disorder associated with excessive food intake (preferred) (obesity (preferred), bulimia nervosa or compulsive eating disorders), metabolic disorders associated with metabolic disorders including obesity, bulimia nervosa, compulsive eating
Activity
Eating-Disorders-Gen.; Anorectic; Tranquilizer; Antidiabetic; Antiarteriosclerotic; Hypotensive; Gynecological; Cardiovascular-Gen.; Antiarthritic; Osteopathic; Dermatological; Antilipemic; Hypnotic; Neuroleptic; Antiaddictive; Antidepressant; Anticonvulsant; Antimanic; Nootropic; Neuroprotective; Antiparkinsonian; Cerebroprotective; Vulnerary; Hypertensive; Respiratory-Gen.; Cardiant; Immunosuppressive; Antirheumatic; Antimigraine; Antiinflammatory; CNS-Gen.; Vasotropic; Gastrointestinal-Gen.; Antiallergic; Antipsoriatic; Antiasthmatic; Thyromimetic; Cytostatic.
Mechanism Of Action
Cannabinoid receptor 1 inhibitor. The ability of (I) to inhibit cannabinoid receptor 1 was tested in a mice using a biological assay. The results showed that (I) exhibited competitive inhibitor constant value of less than 100 nM.
Inventions: 300 out of 300 rows. vmns: 4 out of 579 rows.
Files
Lists and queries
Services
Projects [jic-project-markush-demoset]

Summary

- IJC database + forms + chemistry features provides good solution for serving up database
- Sometimes custom processes will be needed to prepare the data
- Administrators do the hard work
- Users reap the benefits