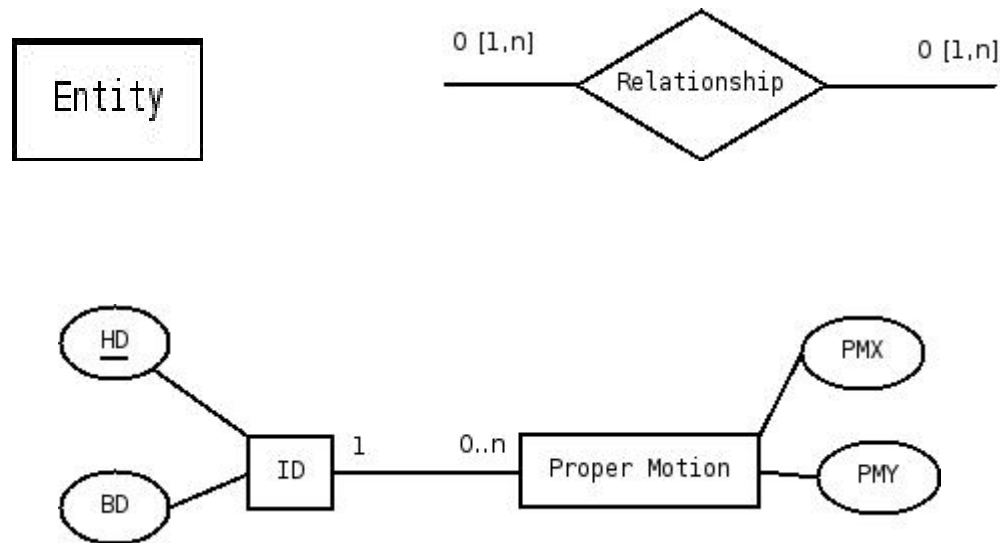


# Data Modeling

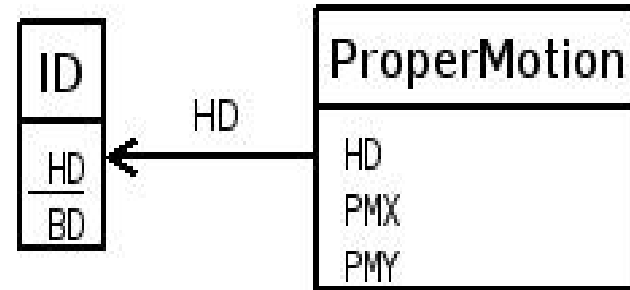
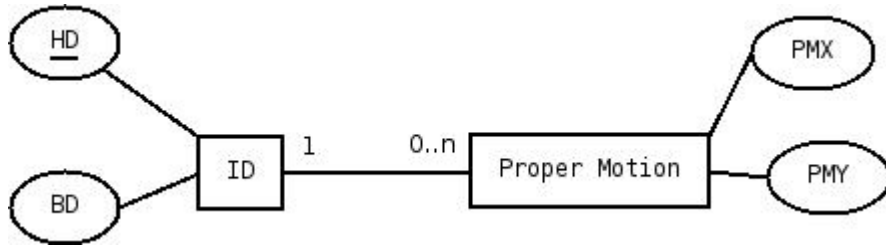
- ER
- SADT
- UML

# Data Modeling: ER

- Entity-relationship diagram
- Entity: attributes
- Relationship: (smth)-(smth)



# Data Modeling: ER



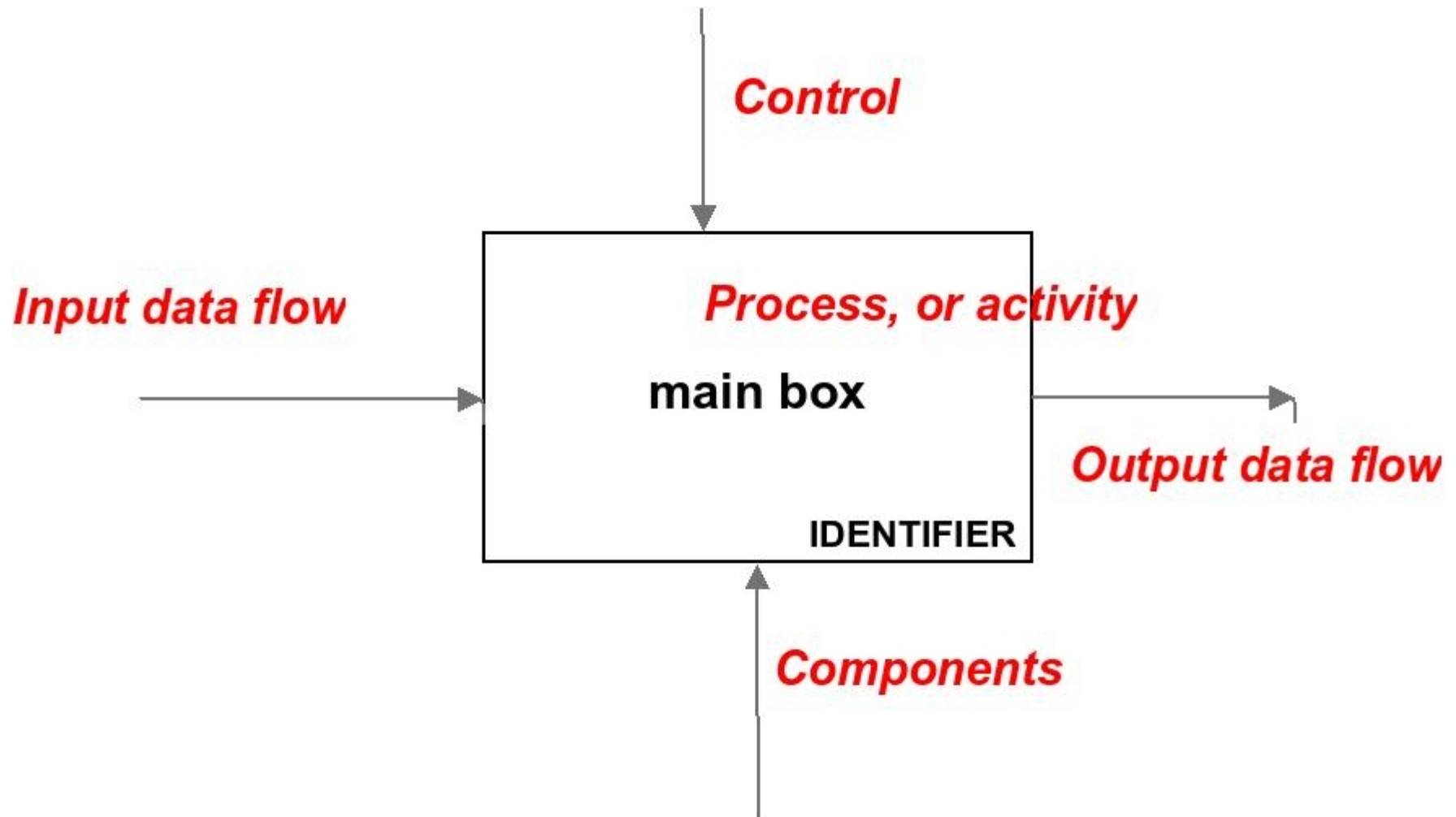
```
CREATE TABLE ID (HD INTEGER NOT NULL,  
                 BD INTEGER,  
                 CONSTRAINT hd_pk PRIMARY KEY (HD))  
CREATE TABLE ProperMotion (HD INTEGER NOT NULL,  
                             PMX DOUBLE,  
                             PMY DOUBLE,  
                             CONSTRAIN hd_fk FOREIGN KEY (HD)  
                             REFERENCES hd_pk)
```

```
SELECT T1.HD, T1.BD, T2.PMX, T2.PMY FROM ID T1, ProperMotion T2 where  
       T2.HD=T1.HD
```

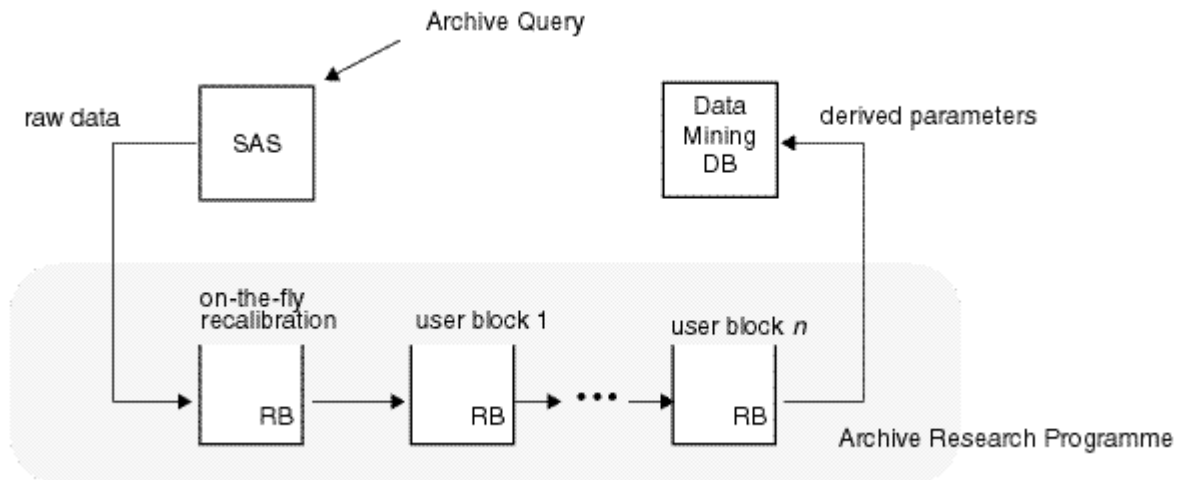
# SADT

- Structured Analysis and Design Technique
- Integration Definition for Function Modeling -0
- Simple 2-elements diagram

# SADT



# SADT

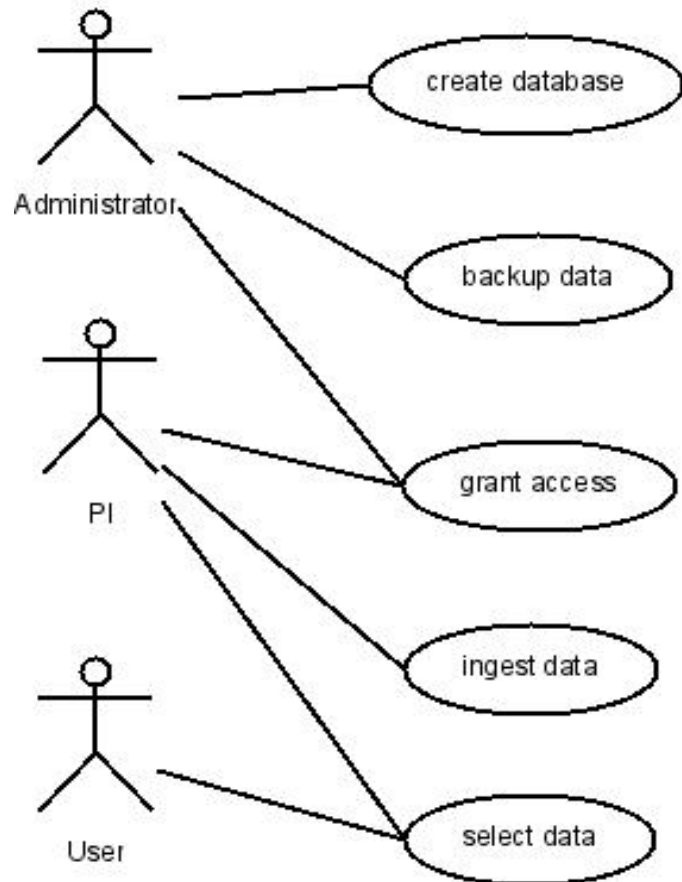
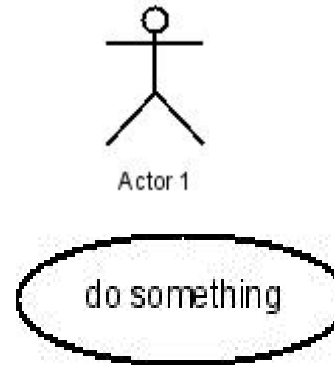


# UML: Unified Modeling Language

- SW design
- Data processing design
- Design → Implementation
- Object-oriented approach
- Database design

# UML: Use Case Diagram

- Actor
- Use Case

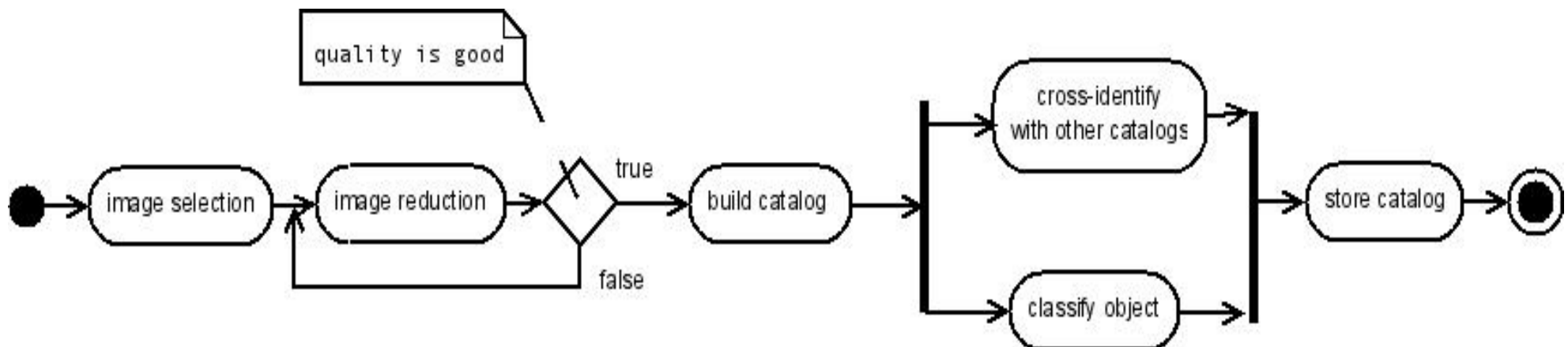
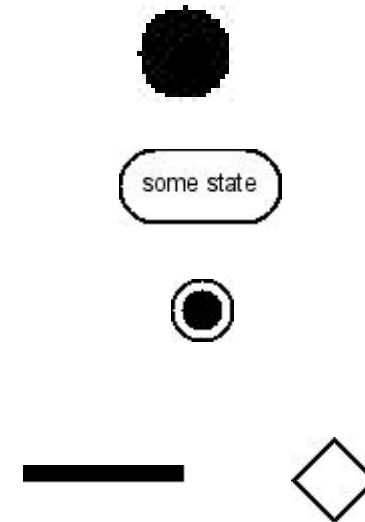


- First step in developing
- Define use cases
- Do not go in details



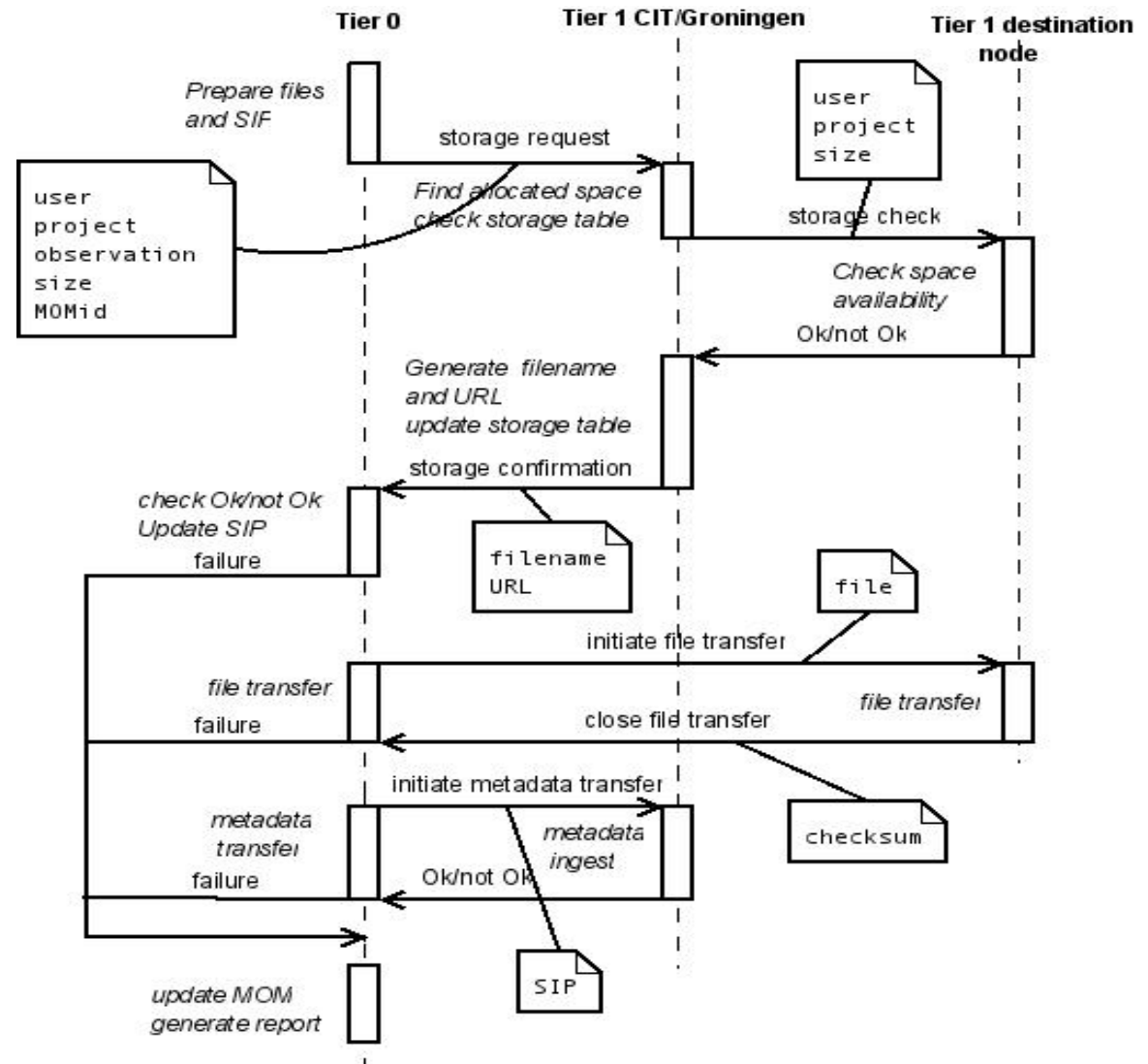
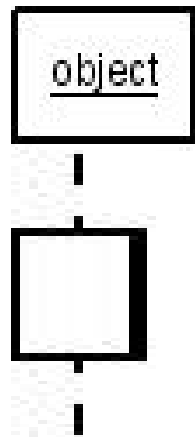
# UML: Activity Diagram

- Initial State
- Activity
- Final (End) State
- Connector (Fork/Union)



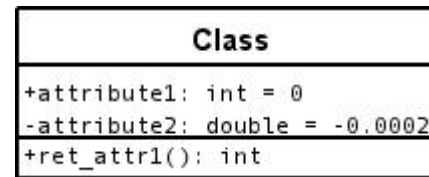
# UML: Interaction Diagram

- Object (participant)
- Lifeline
- Message

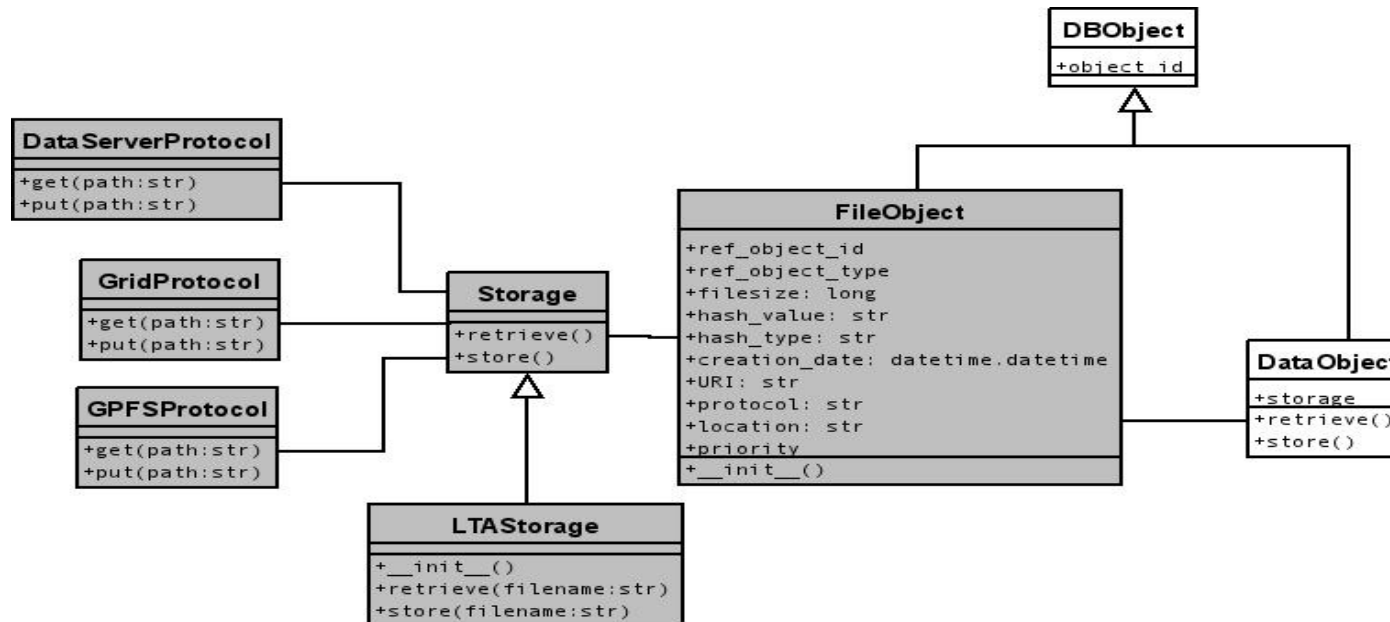


# UML: Class Diagram

- Class
- Object
- Relationship



object.attribute



# UML: diagrams

- Package diagram
- Component diagram
- Deployment diagram
- Statechart diagram

# Data Modeling: UML

diagram of classes

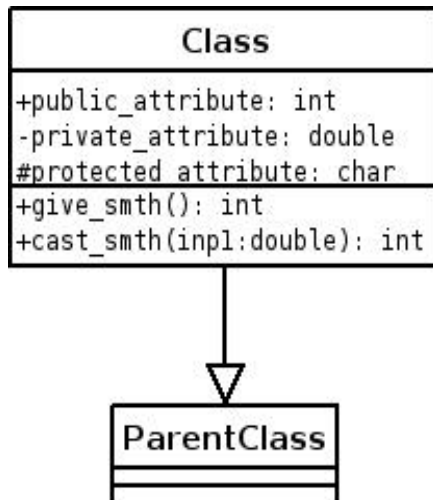
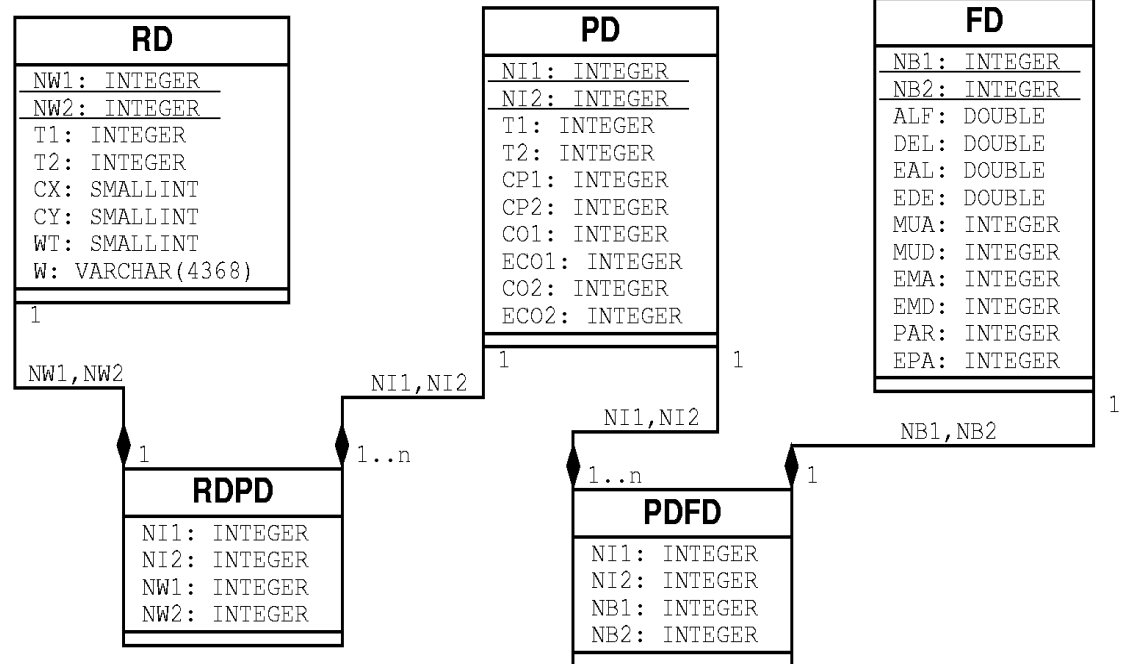
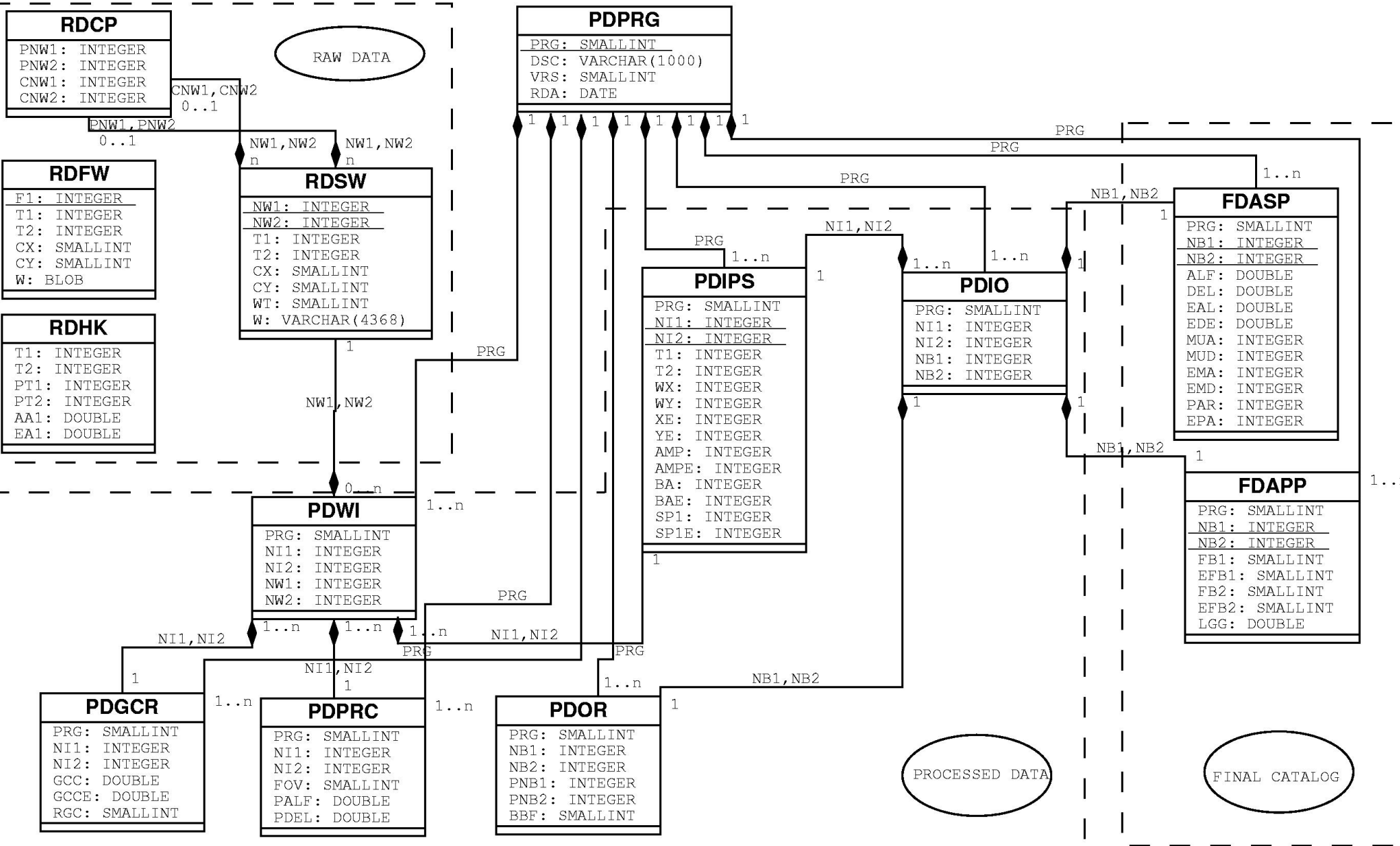


diagram of tables



- the same primitives for another purposes
- data processing

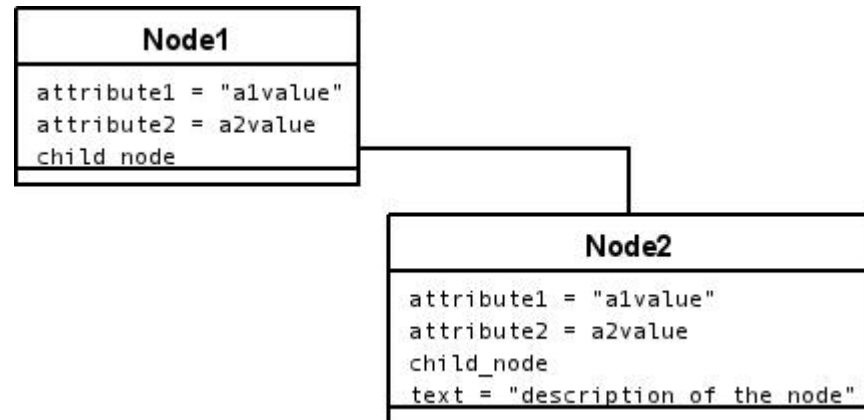


# XML

- Extensible Markup Language
- Self-describing format
- Standard Generalized Markup Language
- HTML
- XML Schema
- XSLT

# XML: structure

- Node (Element)
- Text
- Attributes



```
- <ROOT xmlns="the reference to the scheme">
  - <NODE1 attribute1="a1value" attribute2="a2value">
    <NODE2 attribute1="a1node2value" attribute2="a2node2value"> Just an example </NODE2>
  </NODE1>
</ROOT>
```



# Python

- Python language [www.python.org](http://www.python.org)
- Object-oriented, modular, API
- language-interpreter
  
- Functions `def`
- Classes `class`

# Python: an example

```
from math import pi, sin, cos, sqrt, asin, atan, atan2
from astro.util.extinction import p2i
```

```
class IDLLibError(Exception):
    pass
```

```
def total(a,b):
    try:
        if(len(a)!=len(b)):
            raise IDLLibError, 'total: the same size for input matrix'
    except:
        raise IDLLibError, 'total: input are two matrix of the same size'
    x=0.0
    for i in range(len(a)):
        x=x+a[i]*b[i]
    return x
```

# Python: an example

```
import MySQLdb
```

```
conn = MySQLdb.connect(host = "localhost",user = "root",passwd = "",  
                        unix_socket="/home/belikov/mysql/mysql.sock",  
                        read_default_file="/home/belikov/mysql/my.cnf")
```

```
cursor = conn.cursor ()
```

```
cursor.execute ("DROP DATABASE PROB")
```

```
cursor.execute ("CREATE DATABASE PROB")
```

```
cursor.execute ("USE PROB")
```

```
cursor.execute ("CREATE TABLE TWOMASS (RA DOUBLE, DECL DOUBLE)")
```

```
cursor.close ()
```

```
conn.close ()
```

# R

- Language for wrapping statistical computing
- Site: [www.r-project.org](http://www.r-project.org)
- Arrays
- Plotting
- Aggregate functions