Data Cartridge Technology
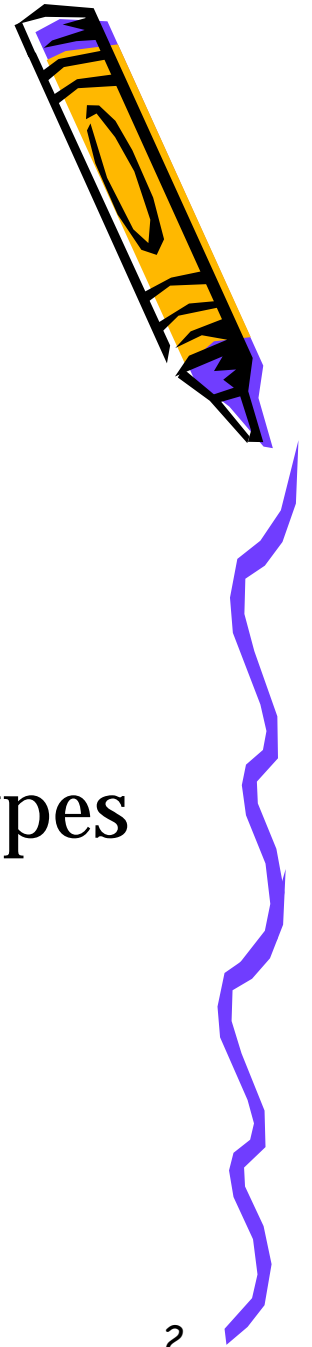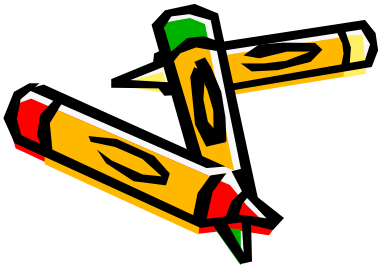
Tharun Kumar Allu

Graduate Student

Department of Computer Science

Extending Databases

Motivation

- New types of data
  - Multimedia, Genomics, Chemistry etc.
- Application Domain Specific data types
- Same level of abstraction

Data Cartridges

- Mechanism to extend Databases.
- A safe, solution-oriented means to package domain-specific data and behavior, and integrate such packages with the server.
  - Ex: A spatial data cartridge may provide comprehensive functionality for a geographic domain such as being able to store spatial data, perform proximity/overlap comparisons on such data, and also integrate spatial data with the server by providing the ability to index on such data.
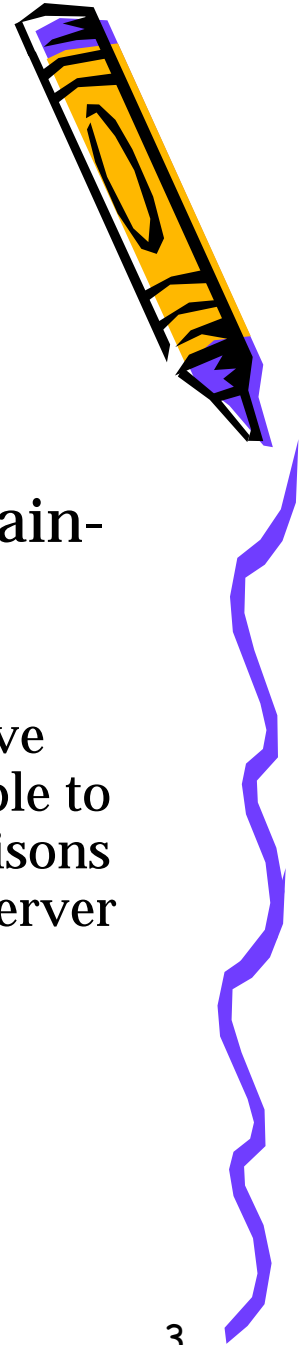
Data Cartridges Contains

- Attribute data that holds object state information. Attributes can be built-in types or other object types.

- Methods that embody the object's behavior. Methods can be simple (such as adding two numbers) or complex (such as computing prices of financial derivatives) and can either be coded in PL/SQL™, in Java™ or in a 3GL like C.
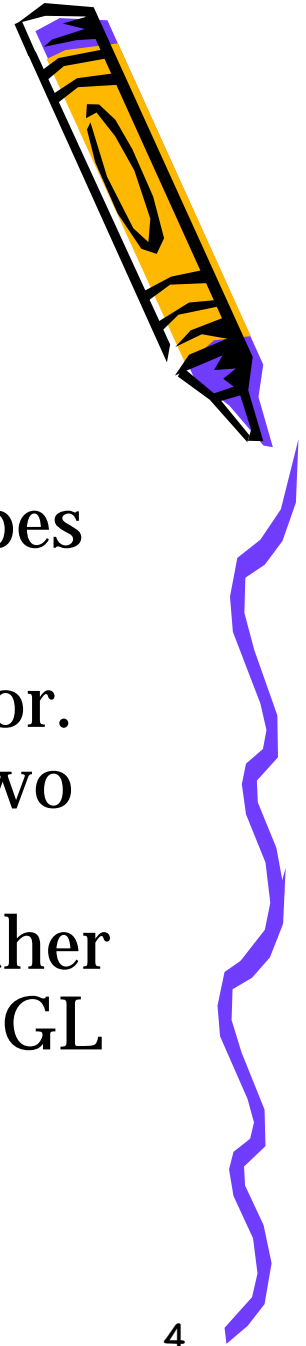
Extensibility Services
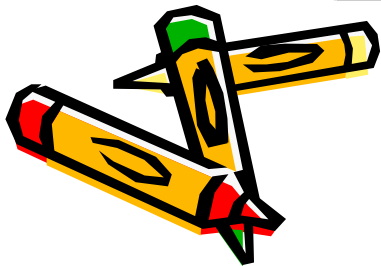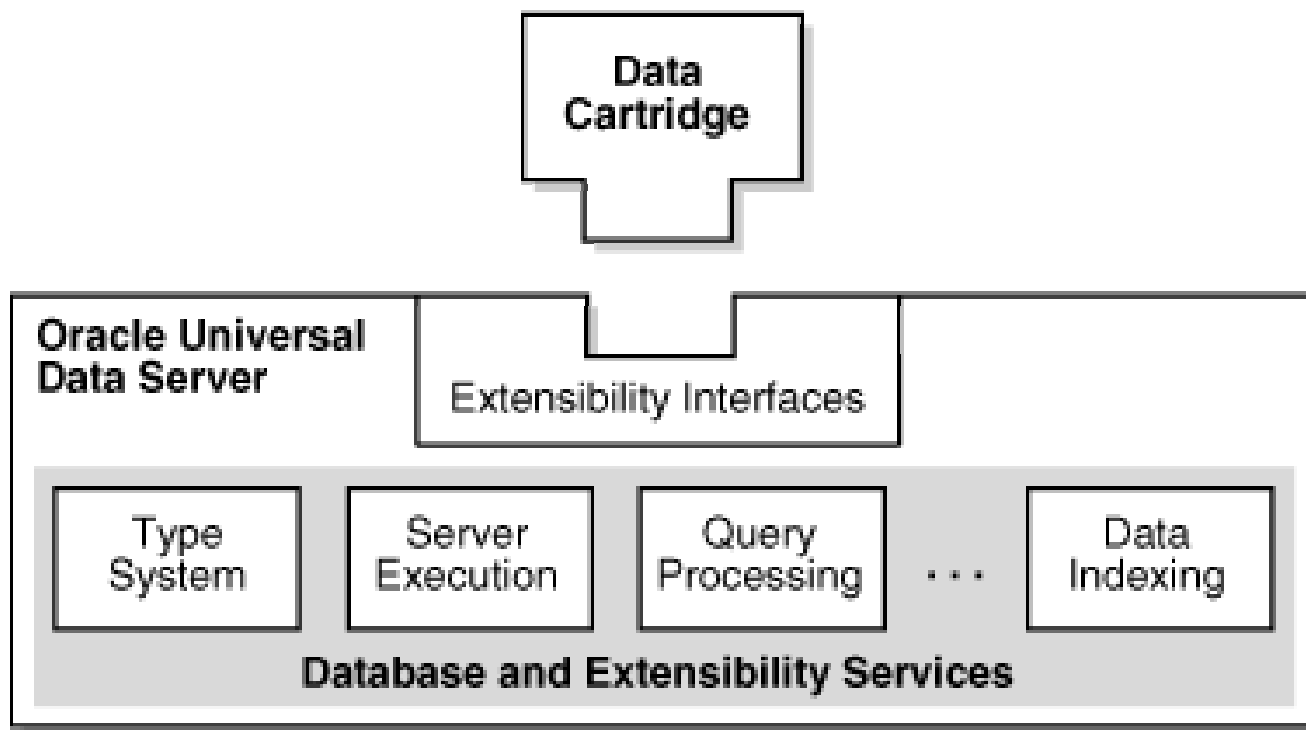
Extensible Type System

- Support for Native types
- Object Types (also called user-defined types or abstract data types)
- Collection Types
  - VARRAY (varying length array)
  - Nested Table (multi-set)
- REF (relationship)
- Large Object Types
  - BLOB (binary large object)
  - CLOB (character large object)
  - BFILE (binary large file object)
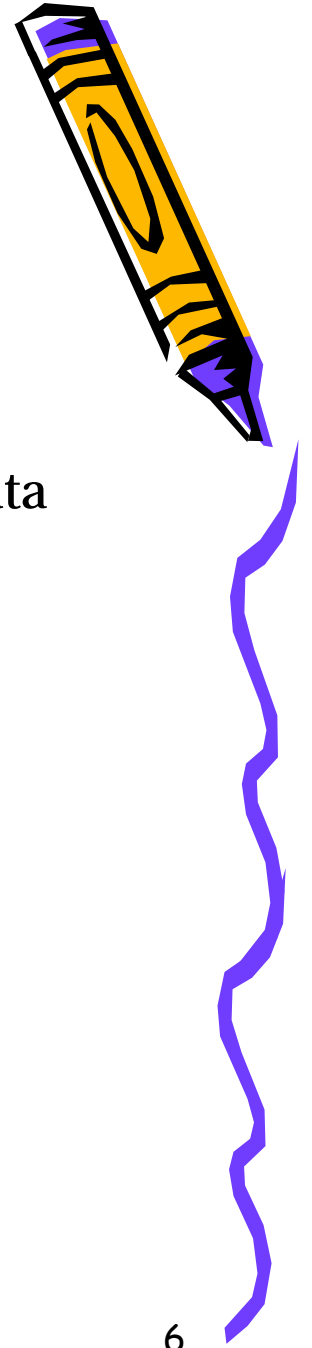
Extensible Server Execution Environment

- What about PL/SQL (Comprehensive procedural language) ?

- Functions like Fast Fourier Transforms, image format conversion, Chemical structure similarity etc. are faster in 'C'

Extensible Server Execution Environment (contd..)

- External routines can callback to Oracle Server using OCI (Oracle Call Interface).

- External programs are executed in a separate address space from the server.

- Java can also be used.
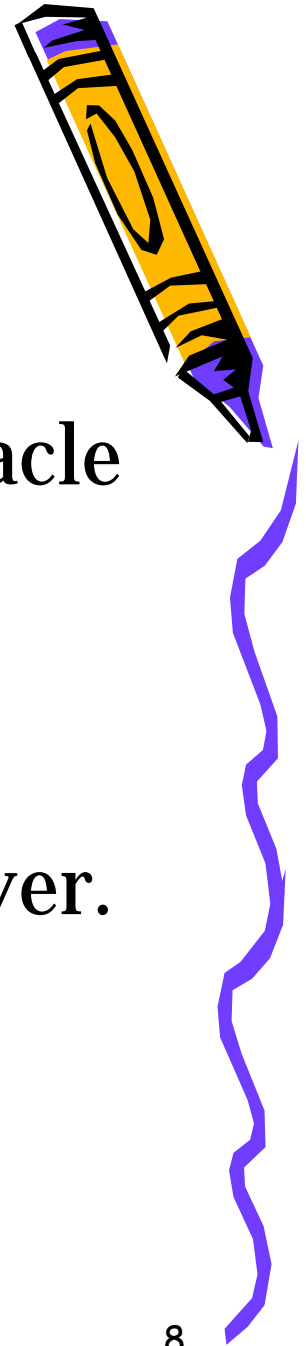
Calling out to External Procedures
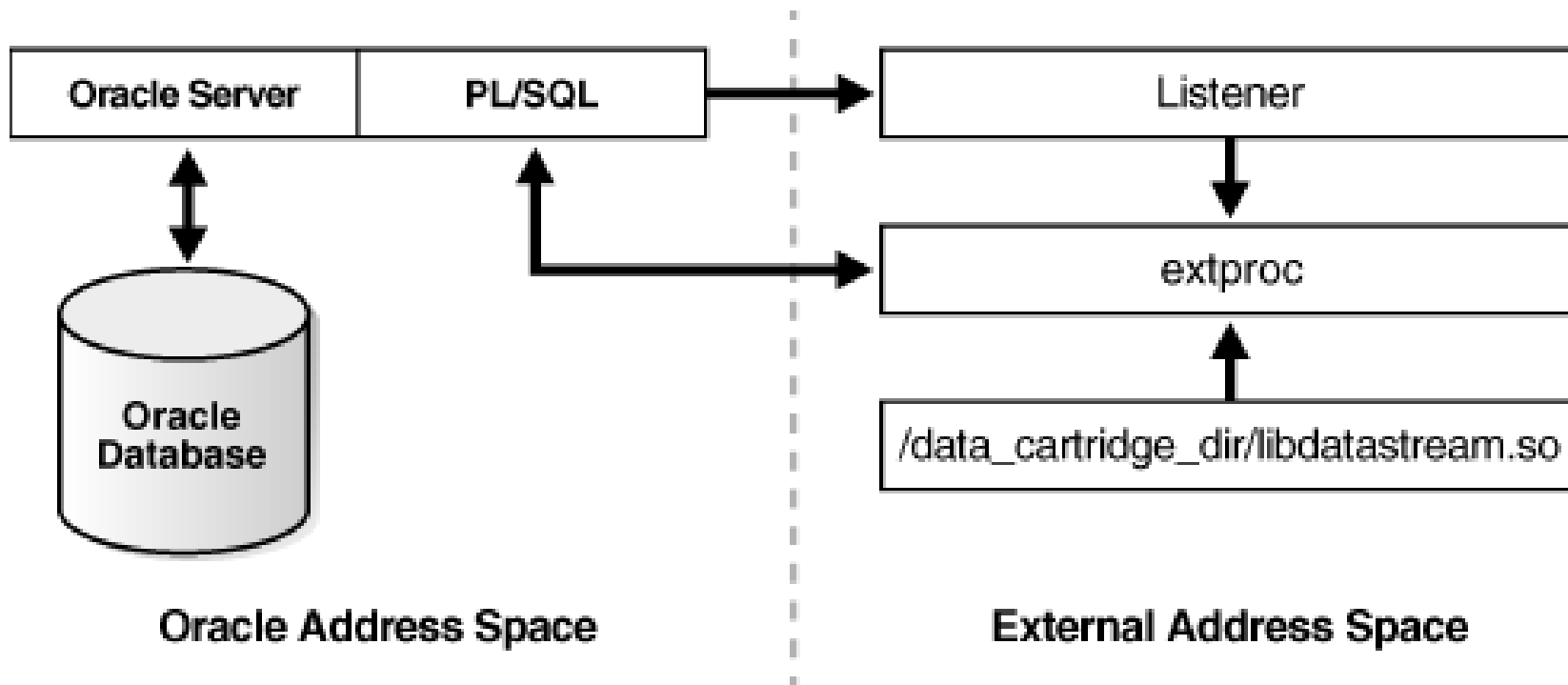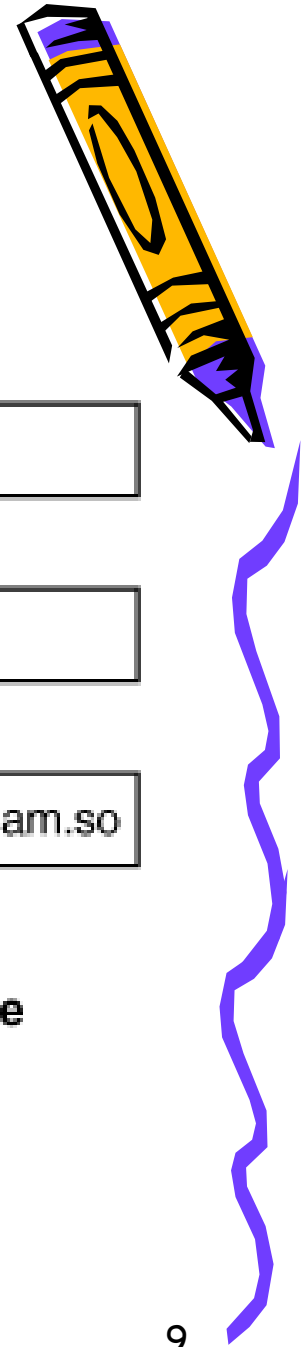
Extensible Indexing

- B Trees, Hashing etc (Number, Strings etc)

- New Data
  - Text, Spacial, image video, audio, Chemical structures, etc (Need context based retrieval)
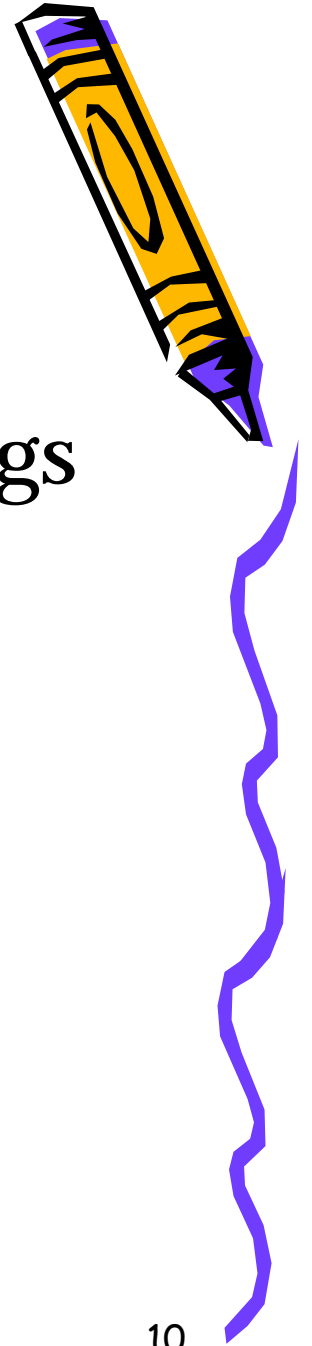
Extensible Indexing (contd…)

- Defines the structure of the domain index as a new indextype

- Stores the index data either inside the database or outside the database

- Manages, retrieves, and uses the index data to evaluate user queries

Extensible Indexing (contd…)

When the database server handles the physical storage of domain indexes, cartridges must be able to:

- Define the format and content of an index. This enables cartridges to define an index structure that can accommodate a complex data object.

- Build, delete, and update a domain index. The cartridge handles building and maintaining the index structures.

- Access and interpret the content of an index. This capability enables the data cartridge to become an integral component of query processing. That is, the content-related clauses for database queries are handled by the data cartridge.
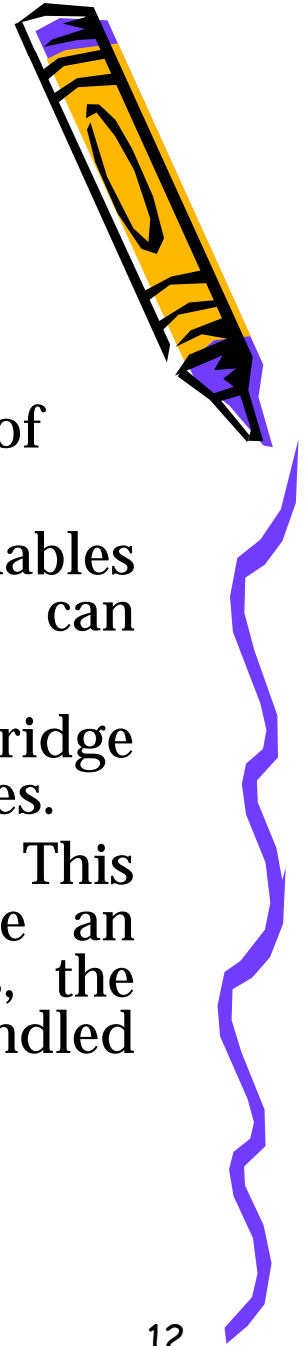
User Defined Operators

Ex:

SELECT * FROM Employees WHERE Contains(resume, 'Perl' AND 'Unix');

It can be used in:

- SELECT command
- Condition of a WHERE clauese.
- ORDER BY or GROUP BY clauses.
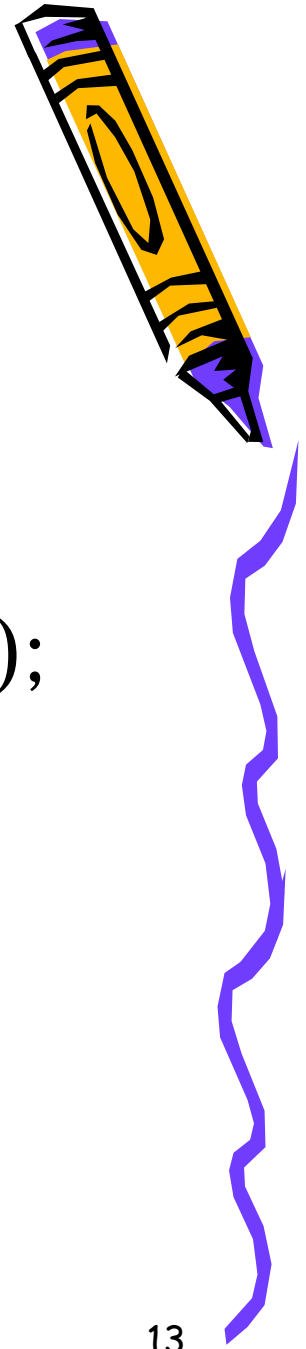
Extensible Optimizer

- ## Statistics
  - Provided to the DB by Cartridge Interface.
- ## Selectivity
  - To determine the optimal join order.
- ## Costs
  - CPU, I/O, Network

Extensibility Interfaces

- DBMS Interfaces
- Data Cartridge Interfaces
- Cartridge Service Interfaces
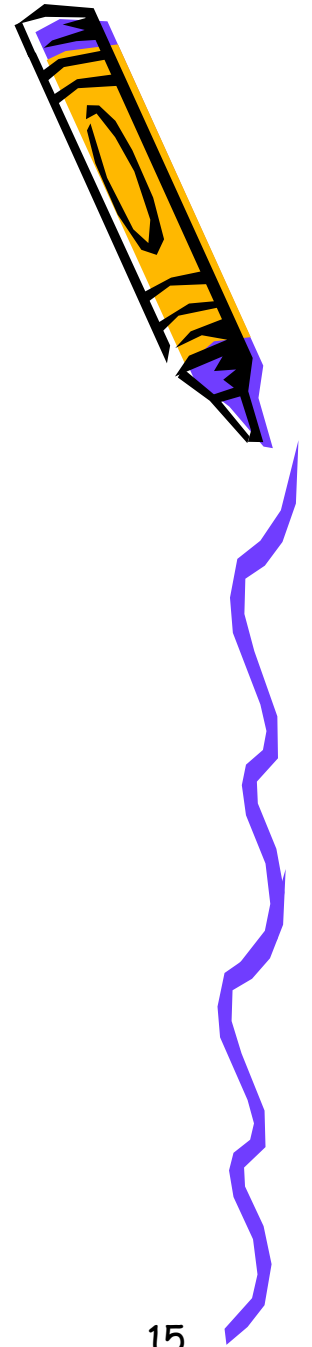
DBMS Interfaces

Extension to SQL or OCI

Ex:

CREATE OR REPLACE TYPE address_t AS OBJECT (

street VARCHAR2(200),

city VARCHAR2(200),

state CHAR(2),

zip VARCHAR2(20)

)

Also Operators, Functions, Indexes
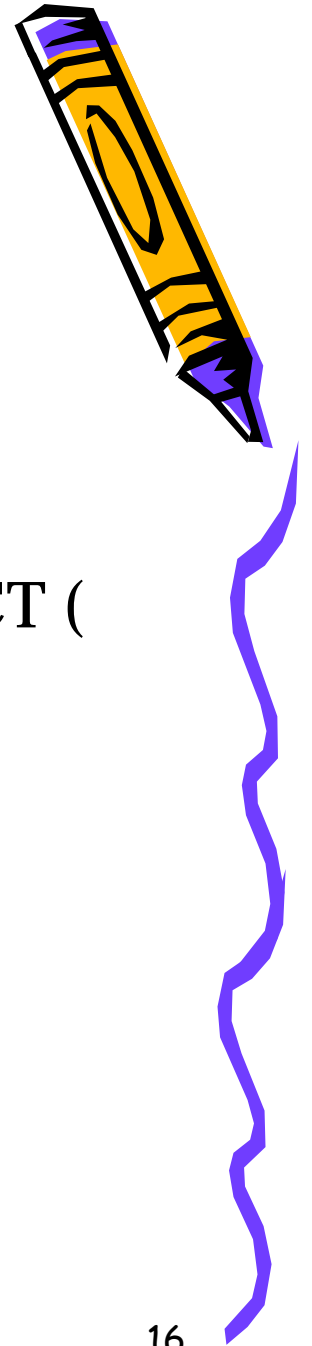
Data Cartridge Interfaces

- DBMS must call cartridge interface for user defined Indexes and user defined Query optimization.

- For ODCIIndex (Oracle Data Cartridge Index) interface refer Oracle Documentation.

Cartridge Service Interface

The interfaces include:

- Memory Management
- File I/O
- Parameter Management
- Internationalization
- Error Reporting
- Context Management

Cartridge Development Process

```
                         ┌──────────────────────┐
                         │   Inventory Domain   │
                         └──────────────────────┘
                                    │
                                    ▼
                         ┌──────────────────────┐
        ┌───────────────▶│   Define key objects │
        │                └──────────────────────┘
        │                           │
        │                           ▼
        │                ┌──────────────────────┐
        │                │  Write SQL and PL/SQL │
        │                │     for object's      │
        │                │  type specification   │
        │                └──────────────────────┘
        │                           │
        │                           ▼
        │                ┌──────────────────────┐
        │                │   Inventory Access    │
        │                │       Methods         │
        │                └──────────────────────┘
        │                           │
        │                           ▼
        │              Yes  ╱◇ Simple ◇╲  No
        │            ┌───────         ───────┐
        │            ▼                       ▼
┌───────────────┐  ┌──────────────────┐  ┌──────────────────┐
│ Package        │◀─│ Build regular    │  │ Define Index     │
│ existing 3GL   │  │ indexes          │  │ types            │
│ code in a DLL  │  └──────────────────┘  └──────────────────┘
└───────────────┘                                   │
        │                                           ▼
        │                    Yes  ╱◇ Multi-domain ◇╲  No
        │                  ┌───────   queries   ───────┐
        │                  ▼                           ▼
        │        ┌──────────────────┐    No  ╱◇ Cost of I/O ◇╲ Yes
        │        │ Implement        │◀───────  only significant ──────┐
        │        │ Extensible       │          factor                 │
        │        │ Optimizer        │                                 ▼
        │        └──────────────────┘                 ┌──────────────────────┐
        │                  │                           │ Use existing optimizer│
        │                  ▼                           └──────────────────────┘
        │        ┌──────────────────┐                             │
        └───────▶│       Test       │◀────────────────────────────┘
                 └──────────────────┘
                           │
                           ▼
                 ┌──────────────────┐
                 │ Installation     │
                 │ Script(s) and    │
                 │ User's Guide     │
                 └──────────────────┘
```
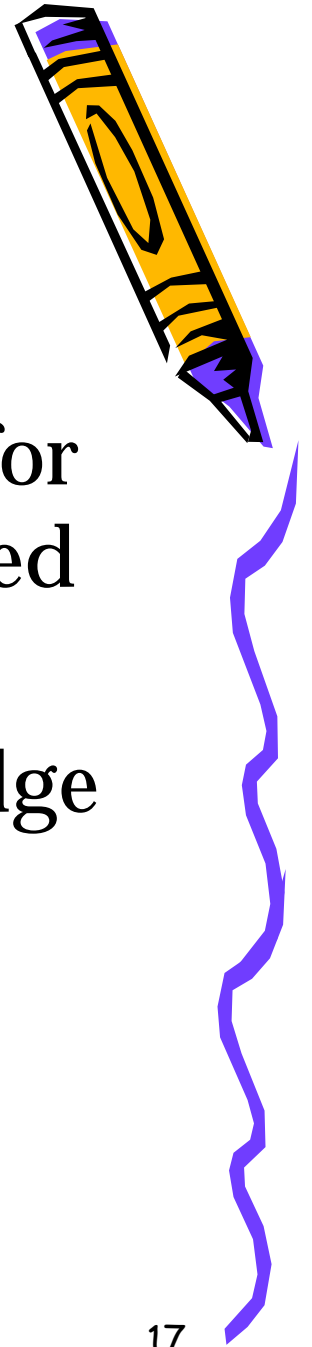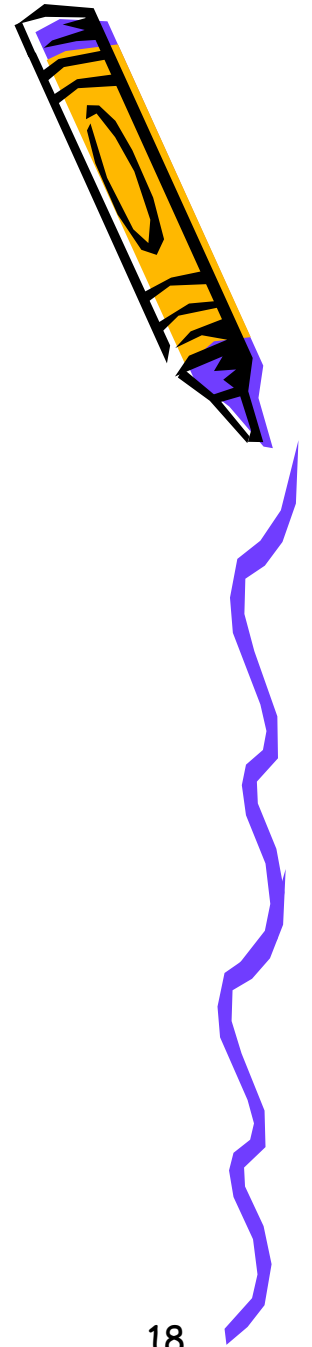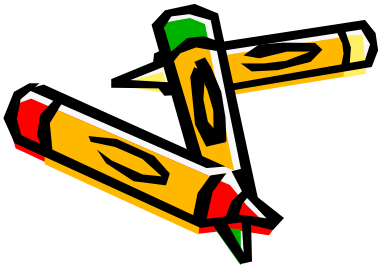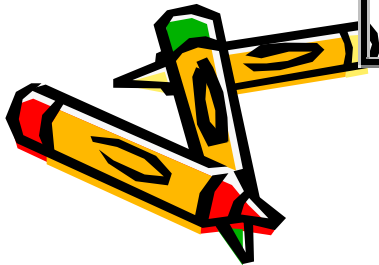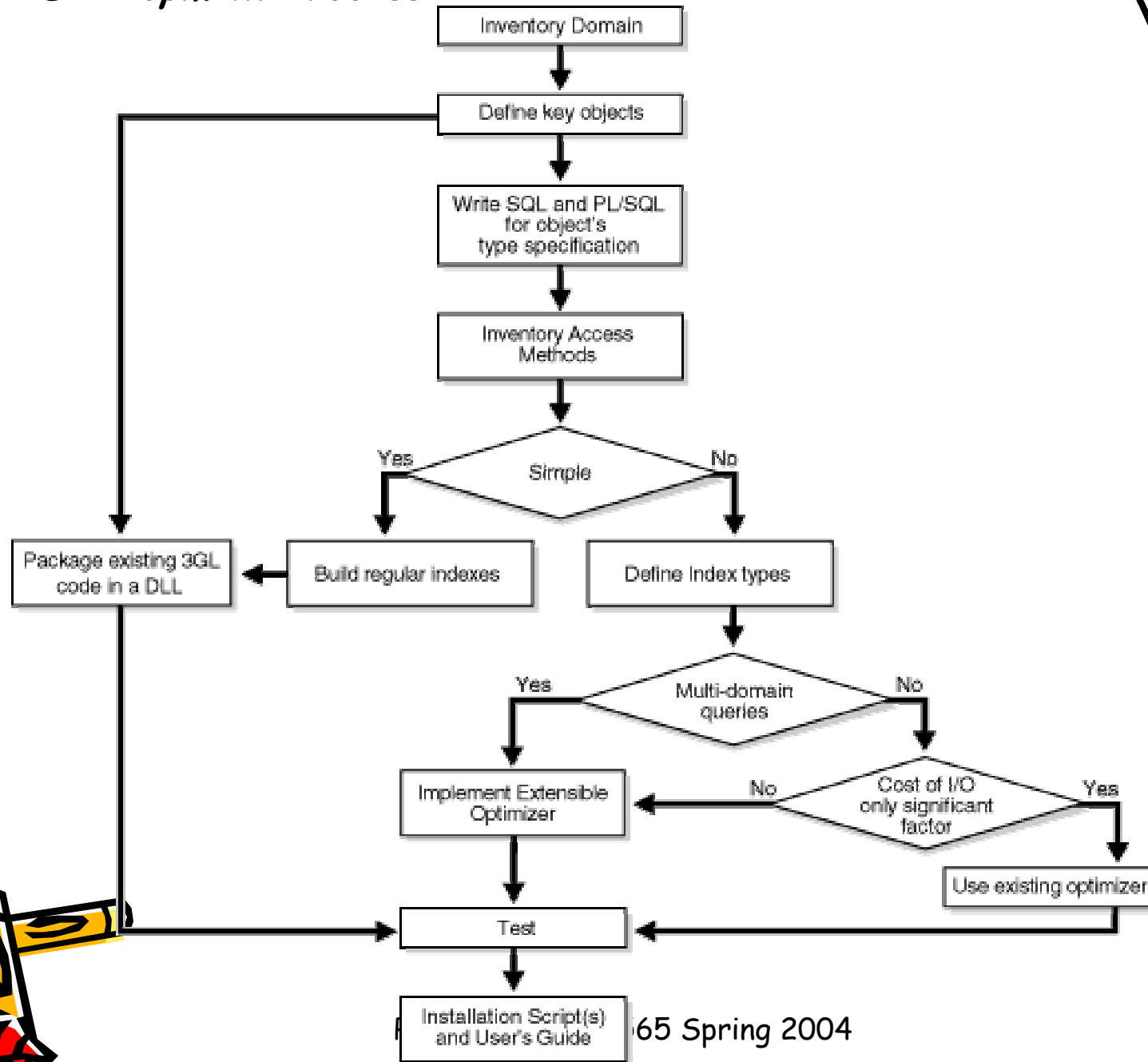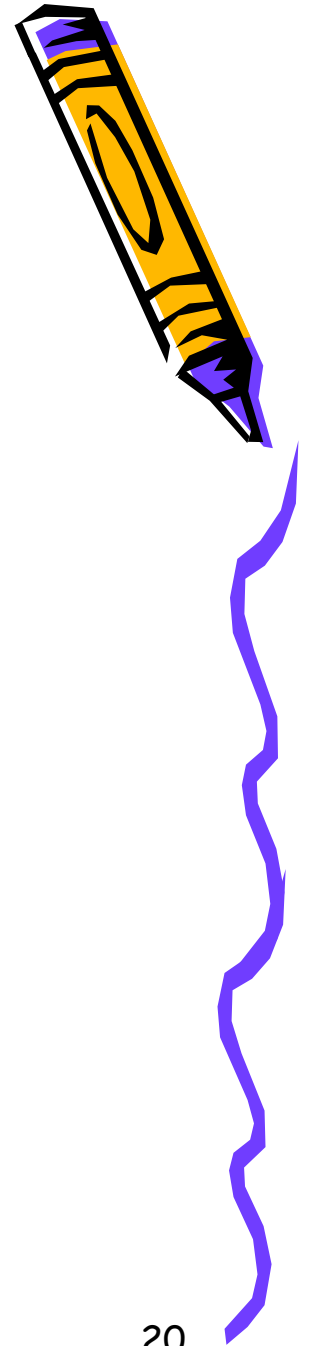
Commercially Available Cartridges

- Daylight DayCart
- Accelrys Accord for Oracle
- MDL ISIS/Direct
- Tripos Auspyx
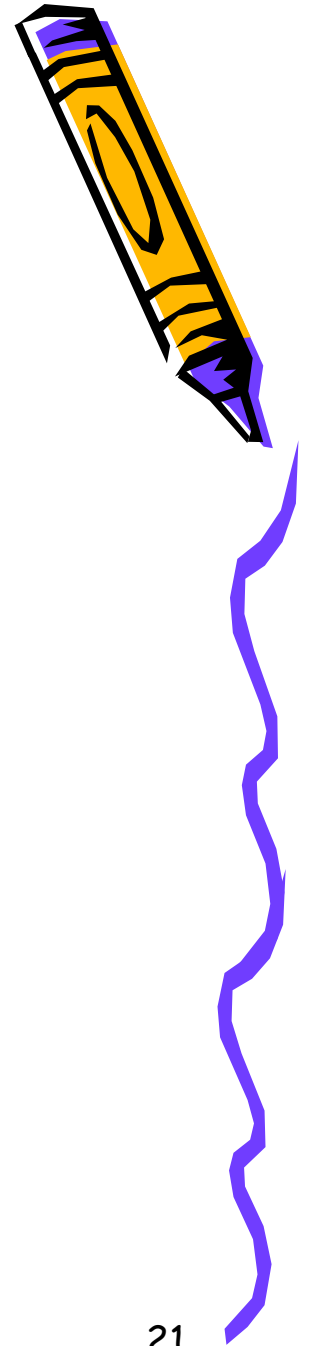- IDBS Chemistry Cartridge
- Cambridgesoft Oracle Cartridge

Performance

```
$ time cansmi < test.smi | wc
SMILES in: 1999; SMILES out: 1999; SMILES changed: 0
So long, baby!
1999 1999 74725
real 0m2.80s
user 0m2.74s
sys 0m0.03s

SQL> select sum(length(smi2cansmi(smiles, 0))) from cansmidemo;
SUM(LENGTH(SMI2CANSMI(SMILES,0)))
---------------------------------
72725
Elapsed: 00:00:02.95
```
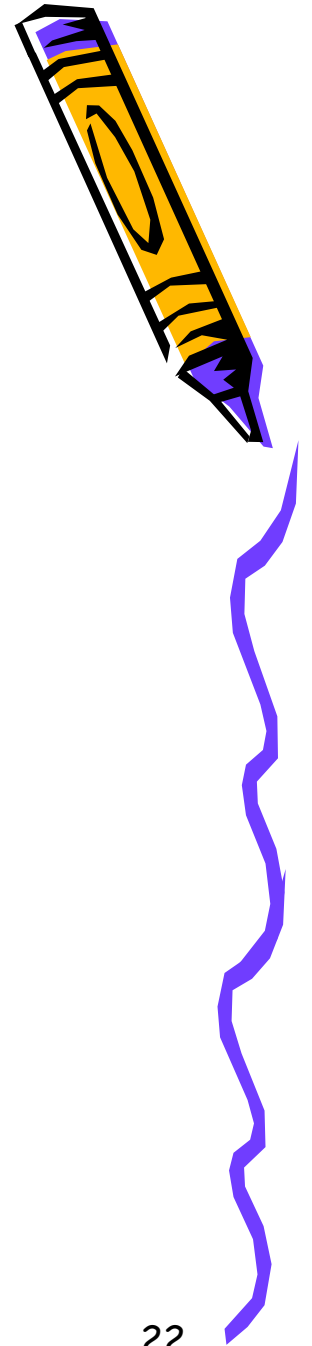
Sizes Inspected

- *small* - 118,611 SMILES
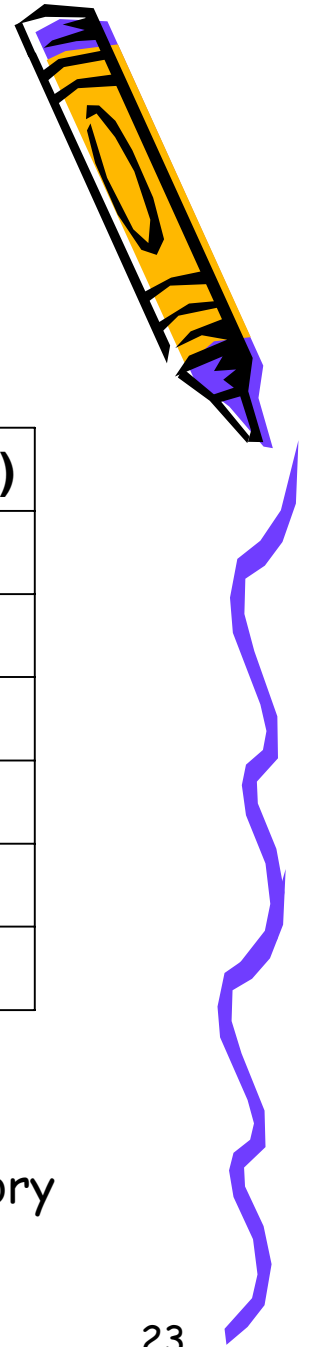- *medium* - 1,097,027 SMILES
- *large* - 5,464,800 SMILES

Index Creation times

Table name	Index type	Column	Creation time (hh:mm:ss)
small	smiles	exact	2:02
small	smiles	blob	11:47*
medium	smiles	exact	22:01
medium	smiles	blob	2:07:00*
large	smiles	exact	1:55:56
large	smiles	blob	11:16:30*

(* - Includes fingerprint generation time)

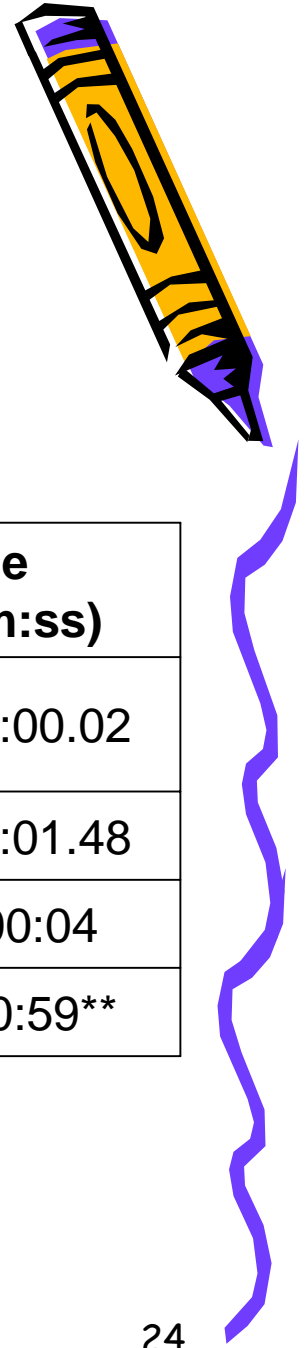Sun Ultra 60 (2x330MHz), with 768 MB of real memory

Query Performance

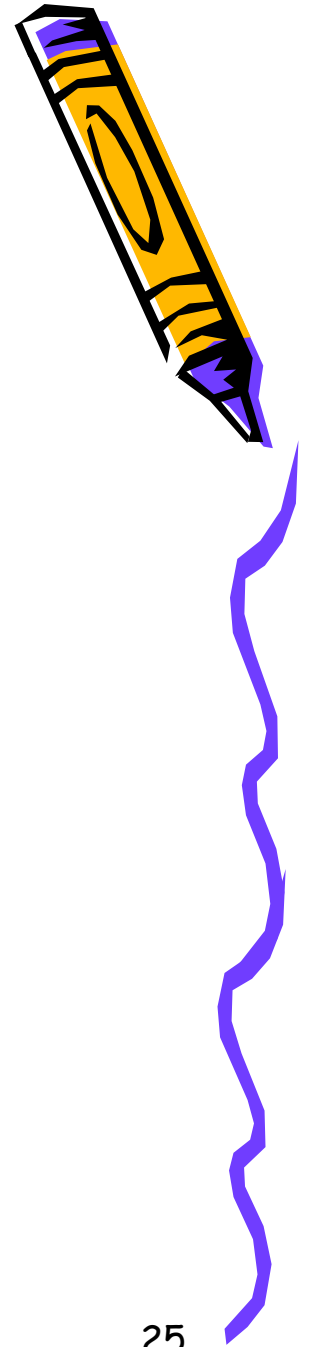| Table name	Query	Hits	Time (mm:ss)
small	contains(smiles, 'OC(=O)C1') = 1	0 (invalid query)	00:00.02
small	contains(smiles, 'OC(=O)CS') = 1	492	00:01.48
medium	contains(smi, 'O1C(=O)CCS1') = 1	2	00:04
large	contains(smiles, 'NCCc1ccc(S)cc1') = 1	2617	00:59**

(** - Disk I/O observed)

Other Databases?

- MySQL
  - Using UDF (User Defined Functions)
- PostgreSQL
  - User defined Functions (C-Language Functions)

References

1. All Your Data: The Oracle Extensibility Architecture, An Oracle Technical White Paper, February 1999 (http://otn.oracle.com/products/oracle8i/pdf/8i_yourdata.pdf)
2. http://otn.oracle.com/products/oracle8i/htdocs/ext.htm
3. http://www.daylight.com/meetings/mug01/Delany/cartridge.html
4. http://www.tripos.com/sciTech/enterpriseInfo/media/opInfoTech/AUSPYXWide%20Release12.18.03.pdf