Ingres
Ingres

Product of ASK Computer Systems, Inc.
   Formerly Ingres, Inc.
   Formerly Relational Technology, Inc.

"Commercial" Ingres
   as opposed to "university" Ingres, a public domain product with few interfaces

Originally "Interactive Graphics and Retrieval system"

Runs on
   DEC - VAX/VMS
   IBM - VM/CMS
   IBM - MVS/XA
   Unix
   IBM PC/Dos

   ... and many others
Components of Ingres

end users

frontends

backends

databases
# Ingres Front-End Programs

<table>
<thead>
<tr>
<th>Ingres/Query</th>
<th>QBF</th>
<th>retrieval/update/entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingres/Reports</td>
<td>RBF</td>
<td>report defn/writing</td>
</tr>
<tr>
<td>Report Writer</td>
<td></td>
<td>customized reports</td>
</tr>
<tr>
<td>Ingres/Graphics</td>
<td>VIGRAPH</td>
<td>business graphics</td>
</tr>
<tr>
<td>Ingres/Forms</td>
<td>VIFRED</td>
<td>form defn/editing</td>
</tr>
<tr>
<td>Ingres/Applications</td>
<td>ABF</td>
<td>application generation</td>
</tr>
<tr>
<td>Terminal monitor</td>
<td>IQUEL/ISQL</td>
<td></td>
</tr>
<tr>
<td>Forms system</td>
<td>FRS</td>
<td>user-developed applications</td>
</tr>
</tbody>
</table>
Ingres Back-End Programs

Back-End Services
- Relational operations
- Data definition
- Data manipulation (SQL/QUEL)
- Security
- Integrity
- Transaction processing
- Concurrency control
- Recovery
- Lock manager
- Embedded QUEL / SQL    (EQUEL / ESQL)
- Business rules / A.I. module

Back-End Mechanisms
- SQL/QUEL parsing
- Query modification
- Query optimization
- Query execution
Ingres Communications Programs

Ingres/Net
Allows Ingres frontends on one machine to interact with the Ingres backend on another

Ingres/Net PC
Allows Ingres frontends to run on a PC, thus the PC acts as a workstation for an Ingres host

Ingres/PCLink
Allows Ingres database information to be downloaded into Lotus 1-2-3, dBase II/dBase III, WordStar, Multiplan, VisiCalc etc. files (and upload)

Ingres/Star
Allows arbitrary table collections from multiple sites to function as a single database. (Join tables across machines)
Ingres Data Definition

Create base tables or end-user views

Destroy base tables or end-user views:

Support for null field values

In Ingres, system information itself can be stored in tables that can be queried
iitables contains a list of all base tables
iicolumns contains a list of all columns
iiviews contains a list of all views
iiindexes contains a list of all indices
and 60 other tables contain other information

Cannot update a table definition
 Need to delete, re-create table

Can update view definition
 append, delete, or replace
Ingres Security

System manager

Database Administrator (DBA)
  has createdb authority
  databases are public by default (can be made private)
  access rights are stored in dbdb
  tables created are global to all users

User
  access to database implies right to create tables
  user-created tables are private to that user

Specific permissions can be granted
QUEL:
  • define permit retrieve on P to Joe where P.city = "London" (Joe can see London parts only.)
  • Other rights: append, replace, delete, all.
  • Remove rights: destroy permit retrieve ...

SQL:
  • grant connect retrieve on P to Joe where ...
  • Other rights: insert, select, update, delete, alter, index, all.
  • Remove rights: revoke connect retrieve ...
Ingres Integrity

Data type constraints
stored in data dictionary

Single variable constraints
QUEL:
• define integrity on S is S.status > 0

SQL:
• create integrity on S is S.status > 0

VIFRED:
• done at the data entry level

SQL/QUEL constraints done by query modification
which is silent

Multi-variable constraints
Done with “A.I.” or Business Rules add-on module

Relation-level, database-level constraints
not implemented
Transaction Processing

A transaction is defined as a *unit of work*

Example: Change a manager's name - two approaches

1. change manager's name field first
   • employee working for manager loses integrity until manager name in row is updated

2. change employee records first
   • lose integrity in employee records until manager's record is updated

Need to update all related fields "as a unit"
be able to back out if anything "fails"

Two methods
a) Begin transaction
   End transaction
   Savepoint x
   Abort
   Abort to x

b) Commit
   Rollback
Concurrency

Locks at the “page” level

Two kinds of locks
  X lock (exclusive) - write lock
  S lock (shared) - read lock

Break deadlock by
  • choosing a "victim"
  • aborting / rolling back the transaction
Embedded SQL

Host Language Interface
Ada, Basic, C, Cobol, Fortran, Pascal, PL/I

Dual-mode Principle
any statement at the terminal can also be used in an application program

Differences in Detail
• how row is returned
• how multiple rows are returned

Host language programs including ESQL statements must be pre-processed
• special prefix EXEC SQL
• can appear within executable statements
• can include references to host variables
• operate on one database at a time
• provide feedback information
• need to worry about ”dereferencing”
ESQL Single Value Example

begin program
EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION;
    title character_string (10);
    hourly_rate float;
EXEC SQL END DECLARE SECTION;
EEC SQL WHENEVER SQLERROR STOP;
EXEC SQL CONNECT educate;
EXEC SQL SELECT job, rate
    INTO :title, :hourly_rate
    FROM emp
    WHERE name = 'jones';
print 'jones', title, hourly_rate;
EXEC SQL DISCONNECT;
end program
Multiple-Row Retrieval ESQL Example

EXEC SQL INCLUDE SQLCA;
EXEC SQL BEGIN DECLARE SECTION;
    name character_string (20);
    hourly float;
EXEC SQL END DECLARE SECTION;
EXEC SQL WHENEVER SQLERROR STOP;
EXEC SQL CONNECT educate;
EXEC SQL DECLARE c1 CURSOR FOR
    SELECT name, rate
    FROM emp
    ORDER BY name;
EXEC SQL OPEN c1;
EXEC SQL WHENEVER NOT FOUND GOTO closecl;
loop indefinitely
    /* the WHENEVER NOT FOUND statement causes
       the loop to be broken as soon as a row is not fetched
    */
    EXEC SQL FETCH c1
        INTO :name, :hourly;
Visual Programming

Ingres / Menu

- forms-based front-end to entire Ingres system
- invoked by
  `ingmenu dbname`
- execute predefined queries, reports, graphs
- invoke frontends QBF, RBF, VIGRAPh, ABF, VIFRED
- enter terminal monitor ISQL, IQUEL
- perform forms-based data definition
Main Menu - VAX

Database: tpsturm

INGRES/MENU

+------------------------------------------------------------------+
¦Tables        ¦Create/examine tables or query/report on table data¦
¦Forms         ¦Create/edit/use forms for customized data access   ¦
¦JoinDefs      ¦Create/edit/use join definitions on multiple tables¦
¦Reports       ¦Create/edit/run reports                            ¦
¦Applications  ¦Create/edit/run 4GL applications                   ¦
¦Queries       ¦Query data using Query-By-Forms or a query language¦
¦              ¦                                                   ¦
+------------------------------------------------------------------+

Place the cursor on your choice and press "Select"

Select(Enter)  Spawn(2)  Help(PF2)  End(PF3)  Quit(PF4)
Table Utility

Forms-based interface for data definition
- create base tables
- destroy base tables
- help on base table definitions
- help on view definitions

Views are created in ISQL / IQUEL

Tables cannot be updated in Ingres - must be deleted and recreated
Ingres Table Utility

TABLES - Tables Catalog

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dept</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>emp</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>proj</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>tasks</td>
<td>tpsturm</td>
<td>table</td>
</tr>
</tbody>
</table>

Place cursor on row and select desired operation from menu.

Create(1)  Destroy(2)  Examine(3)  Query(4)  Report(5)  >
Ingres Table Creation

TABLES - Create a Table

Enter the name of the new table:

Enter the column specifications for the new table:

| Column Name | Data Type | Key # | Nulls | Defaults |
+-------------+-----------+-------+-------+----------|
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |
|             |           |       |       |          |

Insert(1)  Delete(2)  Blank(3)  Move(4)  GetTableDef(5)  >
# Ingres Table Information

**TABLES - Examine a Table**

Information on Table emp

<table>
<thead>
<tr>
<th>Owner: tpsturm</th>
<th>Table Type: user table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Width: 68</td>
<td>Storage Structure: btree unique</td>
</tr>
<tr>
<td>Columns: 7</td>
<td>Pages/Overflow: 4/0</td>
</tr>
<tr>
<td>Rows: 14</td>
<td>Journaling: disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Key #</th>
<th>Nulls</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>c10</td>
<td>1</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>job</td>
<td>c12</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>mgr</td>
<td>c10</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>hired</td>
<td>date</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>rate</td>
<td>money</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>bonus</td>
<td>money</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>deptno</td>
<td>i2</td>
<td></td>
<td>yes</td>
<td>n/a</td>
</tr>
</tbody>
</table>

NewTable(1)  Help(PF2)  End(PF3)
ISQL and IQUEL

Forms-based interfaces
- screen "editor" for commands
- execution window

Development tool
- can save queries to a file (from editor window)
- can scroll through output (from execution window)
- can terminate execution of a query after examination of the first screen of results
Query Selection Menu

Database: tpsturm2

INGRES/MENU
Queries

+-------------------------------------------------------------+
|SQL | Enter interactive SQL statements                           |
|QUEL | Enter interactive QUEL statements                        |
|QBF | Use Query-By-Forms to add/delete/change/view data         |
|     |                                                             |
|     |                                                             |
|     |                                                             |
|     |                                                             |
+-------------------------------------------------------------+

Place the cursor on your choice and press "Select"

Select(Enter) Cancel(.) Spawn(3) Help(PF2)
ISQL Command Window

Enter SQL Statements

+--------------------------------------------------------------------------+
| select name, job, mgr                                                     |
| from emp;                                                                 |
+--------------------------------------------------------------------------+

Go(Enter) Resume(2) Complete(3) Blank(4) Edit(5) File(6) >
### ISQL Execution Window

Start of Output

```
1> select name, job, mgr
2> from emp

+----------------------------------+
¦name      ¦job         ¦mgr       ¦
+----------+------------+----------¦
¦allen     ¦programmer  ¦barger    ¦
¦barger    ¦supervisor  ¦turner    ¦
¦jones     ¦programmer  ¦radl      ¦
¦king      ¦clerk       ¦barger    ¦
¦martin    ¦programmer  ¦barger    ¦
¦olson     ¦analyst     ¦radl      ¦
¦pearson   ¦programmer  ¦radl      ¦
¦radl      ¦supervisor  ¦turner    ¦
¦rogers    ¦programmer  ¦barger    ¦
¦smith     ¦programmer  ¦barger    ¦
¦sturm     ¦clerk       ¦radl      ¦
¦thomas    ¦analyst     ¦barger    ¦
¦turner    ¦supervisor  ¦          ¦
¦vogel     ¦consultant  ¦turner    ¦
+----------------------------------+
```

---

Print(1)  File(2)  Help(PF2)  End(PF3)
Ingres / Query

QBF - Query By Forms

No knowledge of SQL or QUEL required

Performs

• retrievals
• updates
• appends

Varied functions of QBF

• Given a table name, constructs and uses a default form for table query or data entry
• Given the name of a pre-defined join, uses a default form for master/detail table(s) data query or entry
• Given a form name, uses that form for data query or entry to the table(s) or pre-defined join(s) associated with the form
QBF Start-Up Screen

QBF - Start-Up Frame

Query-By-Forms (QBF) is an interactive interface for adding, deleting, changing and viewing data in your database.

You may get a catalog of QBFNames, JoinDefs, or Tables in your database to serve as a basis for editing and viewing. In summary, they are:

- **QBFName** - a name that combines a joindef or table name with a form name. QBFNames are created in VIPRED.
- **JoinDef** - a stored specification of joins between tables, and rules governing how to update data in QBF.
- **Table** - a table or view in your database.

From each of the catalogs, you can browse names, and choose one to serve as a basis for editing or viewing.

Additional utility commands on this menu are:

- **Help** - displays help on QBF and your terminal key functions.
- **Quit** - leaves QBF.

QBFNames(1)  JoinDefs(2)  Tables(3)  Help(PF2)  Quit(PF4)  :
QBF Table Selection Screen

QBF - Tables Catalog

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dept</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>emp</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>proj</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>tasks</td>
<td>tpsturm</td>
<td>table</td>
</tr>
</tbody>
</table>

Place cursor on row and select desired operation from menu.

Create(1)  Destroy(2)  Examine(3)  Go(Enter)  Help(PF2)  > :
QBF Query Type Selection Menu

QBF - Tables Catalog

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dept</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>emp</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>proj</td>
<td>tpsturm</td>
<td>table</td>
</tr>
<tr>
<td>tasks</td>
<td>tpsturm</td>
<td>table</td>
</tr>
</tbody>
</table>

Place cursor on row and select desired operation from menu.

SimpleFields(1)  TableField(2)  Help(PF2)  End(PF3)  :

Copyright © 1971-2002 Thomas P. Sturm  Ingres  28
Query Options Screen

QBF - Execution Phase

Append(1) Retrieve(2) Update(3) Help(PF2) End(PF3) > :
### Query Formulation Screen

**EMP Table**

<table>
<thead>
<tr>
<th>Name: *n</th>
<th>Job: programmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mgr:</td>
<td>Hired:</td>
</tr>
<tr>
<td>Rate:</td>
<td>Bonus:</td>
</tr>
<tr>
<td>Deptno:</td>
<td></td>
</tr>
</tbody>
</table>

Go(Enter)  Blank(2)  LastQuery(3)  Order(4)  ListChoices(9)  >
# First Retrieved Screen

**EMP Table**

<table>
<thead>
<tr>
<th>Name: allen</th>
<th>Job: programmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mgr: barger</td>
<td>Hired: 09-jun-1981</td>
</tr>
<tr>
<td>Rate: $ 30.00</td>
<td>Bonus:</td>
</tr>
<tr>
<td>Dept:no: 402</td>
<td></td>
</tr>
</tbody>
</table>

Next (Enter)  Query(2)  Help(PF2)  End(PF3)
Second Retrieved Screen

EMP Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Job</th>
</tr>
</thead>
<tbody>
<tr>
<td>martin</td>
<td>programmer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mgr</th>
<th>Hired</th>
</tr>
</thead>
<tbody>
<tr>
<td>barger</td>
<td>09-nov-1981</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rate</th>
<th>Bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>$25.00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deptno</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>402</td>
<td></td>
</tr>
</tbody>
</table>

Next (Enter)  Query (2)  Help (PF2)  End (PF3)
Join Definition

Master / master join definition (or detail / detail)
  • ordinary natural join (yields one table)

Master / detail join definition
  • left outer natural join
    - outer join on master (all master occurrences)
    - natural join on detail (matching master must exist)
  • presented in master/detail form
    - master appears in "row" form - one per screen
    - detail appears in "column" form, scrolling on same screen as corresponding master
QBF Initial Screen
QBF JoinDefs Catalog
QBF JoinDefs Definition
QBF Menu to Select Execution Mode
QBF Query Specification Screen
QBF Query Results
(Master / Detail Form)
Ingres / Forms

VIFRED - Visual Forms Editor

Create forms for use with
- QBF (usually defaults are adequate)
- user-developed applications

Can either
- start from scratch
- edit existing form
- edit default form from QBF

Once created, can be used with
- any number of applications
- any number of tables
VIFRED Form Fields

In QBF, fixed relationship between form fields and database fields

In VIFRED
- each form has a name
- each form field has an internal name field
- each form field also has a title known to the user

Can establish
- position on the screen
- tabbing order for data entry
- "trim" or printed text on the screen
- titles
- data window
- attributes
  - internal name
  - display attributes
  - validation checks
  - error messages
VIFRED Initial Screen
VIFRED Forms Catalog
VIFRED Prototype Form
Description of Components in a VIFRED Table
VIFRED Attributes for an Entry / Display Field
Ingres / Reports

RBF - Report By Forms

Report Writer Facility
- content / layout controlled by report specifications
- RBF creates report specifications

Process:
1. Create report specifications (RBF)
2. Compile specifications (SReport)
3. Execute report on demand (Report)

Report writer different that VIFRED
- output only
- supports control breaks

Usually create a VIEW with. joins, calculations, and all report entries prior to invoking RBF
ISQL Window to Create Report View
RBF Initial Screen
RBF Catalog of Reports
RBF Form Editing Screen
RBF Sorting Order / Break Table
RBF Column Break Specifications
Initial Screen for Running the Report