The *display2D* operator of the relational algebra 1

I Flat Relations

1. Displaying text

```plaintext
domain x intg;
domain y intg;
domain textstring strg;
relation Text(x, y, textstring) <-{
  (5000, 4000, "(5000, 4000)") ,
  (2000, 4000, "(2000, 4000)") ,
  (5000, 3000, "(5000, 3000)")};
NewText <- display2D( ) Text;
```

T. H. Merrett ©07/2
The display2D operator of the relational algebra

I Flat Relations

2. Displaying a Set of Points

domain x intg;
domain y intg;
relation Points (x, y) <-{
    (5000, 4000),
    (2000, 4000),
    (5000, 3000)};
NewPoints <- display2D( ) Points;
The *display2D* operator of the relational algebra

### 1 Flat Relations

3. Displaying a Set of Labelled Points

```plaintext
domain label strg;
domain lc intg;
relation LabelledPoints (x, y, lc, label) <-{
    (5000, 4000, 0, "(5000,4000)")
    (2000, 4000, 0, "(2000, 4000)")
    (5000, 3000, 0, "(5000, 3000)")
};
NewLabelledPoints <- display2D( ) LabelledPoints;
```

T. H. Merrett

©07/2
The *display2D* operator of the relational algebra 4

I Flat Relations

4. Displaying a Set of Lines

domain x1 intg;
domain y1 intg;
domain x2 intg;
domain y2 intg;
relation Lines(x1, y1, x2, y2) <- {
  (1363, 3013, 2942, 3010),
  (2942, 3010, 3426, 1508),
  (3426, 1508, 2148, 583),
  (2148, 583, 873, 1514),
  (873,1514, 1363, 3013)};
NewLines <- display2D( ) Lines;
The \textit{display2D} operator of the relational algebra 5

I Flat Relations

4. Displaying a Set of Lines, Part 2

A graph is a set of lines:

\begin{verbatim}
relation f(x,y) <- {(-2.0,-5.0),(-1.5,-2.5),
(-1.0, 0.0),(-0.5, 0.625), (0.0, 1.0),
(0.5, 1.875),(1.0, 4.0),(1.5, 6.875),(2.0,17.0)};
let x1 be fun succ of x order x;
let y1 be fun succ of y order x;
newf <- display2D() where x1>x in f;
\end{verbatim}
The *display2D* operator of the relational algebra

I Flat Relations

5. Displaying a Set of Labelled Lines

relation LabelledLines(x1, y1, x2, y2, lc, label) <-{
(1363, 3013, 2942, 3010, 0, "line1"),
(2942, 3010, 3426, 1508, 0, "line2"),
(3426, 1508, 2148, 583, 0, "line3"),
(2148, 583, 873, 1514, 0, "line4"),
(873, 1514, 1363, 3013, 0, "line5")};
NewLabelledLines <- display2D( ) LabelledLines;
The \textit{display2D} operator of the relational algebra 7

\section*{I Flat Relations}

6. Displaying a Set of Triangles

\begin{verbatim}
domain x1, y1, x2, y2, x3, y3, lc, fc, fp intg;
relation Triangle(x1, y1, x2, y2, x3, y3, lc, fc, fp)<-{
    (5000, 4000, 2000, 4000, 5000, 3000, 1, 6, 50),
    (3000, 1000, 5000, 1000, 5000, 2500, 6, 1, 50)};
NewTriangle <- display2D( ) Triangle;
\end{verbatim}
The *display2D* operator of the relational algebra

## I Flat Relations

### 7. Displaying a Set of Labelled Triangles

relation LabelledTriangle
(x1, y1, x2, y2, x3, y3, lc, fc, fp, label)<- {
    (5000, 4000, 2000, 4000, 5000, 3000, 1, 6, 50, "Tri1"),
    (3000, 1000, 5000, 1000, 5000, 2500, 6, 1, 50, "Tri2"));
NewLabelledTriangle <- display2D( ) LabelledTriangle;
The display2D operator of the relational algebra

I Flat Relations

8. Displaying a Sequenced Polyline

relation Polyline(x, y, sq) <-{
  (1363, 3013, 1),
  (2942, 3010, 2),
  (3426, 1508, 3),
  (2148, 583, 4),
  (873, 1514, 5),
  (1363, 3013, 6)};
NewPolyline <- display2D( ) Polyline;
The \textit{display2D} operator of the relational algebra 10

\section*{I Flat Relations}

9. Displaying a Sequenced Polyline with Labelled Vertices

\begin{verbatim}
relation LabelledVertexPolyline(x, y, sq, lc, label) <-{
  (1363, 3013, 1, 0, "(1363, 3013)"),
  (2942, 3010, 2, 0, "(2942, 3010)"),
  (3426, 1508, 3, 0, "(3426, 1508)"),
  (2148, 583, 4, 0, "(2148, 583)"),
  (873, 1514, 5, 0, "(873, 1514)"),
  (1363, 3013, 6, 0, "(1363, 3013)");
NewLabelledVertexPolyline <- display2D( ) LabelledVertexPolyline;
\end{verbatim}
The \textit{display2D} operator of the relational algebra

\section*{II Nested Relations}

1. Displaying a Sequenced Polyline with a Label

\begin{verbatim}
domain lc intg;
domain Polyline (x, y, sq);
relation NestedPolyline ( label, lc, Polyline)<- {
    ("P1", 0, {(1363, 3013, 1), (2942, 3010, 2),
               (3426, 1508, 3), (2148, 583, 4),
               (873, 1514, 5), (1363, 3013, 6)})};
NewNestedPolyline <- display2D( ) NestedPolyline;
\end{verbatim}
II Nested Relations

2. Displaying Several Polylines or a Combination of Different Shapes

domain Polyline (x, y, sq);
domain NestedPolyline (label, lc, Polyline);
domain LabelledTriangle (x1, y1, x2, y2, x3, y3, lc, fc, fp, label);
relation Graph (NestedPolyline, LabelledTriangle)<-{
    {("P1", 0, {(1363, 3013, 1), (2942, 3010, 2), (3426, 1508, 3),
        (2148, 583, 4), (873, 1514, 5), (1363, 3013, 6)})},
    {(5000, 4000, 2000, 4000, 5000, 3000, 1, 6, 50, "Tri1"),
        (3000, 1000, 5000, 1000, 5000, 2500, 6, 1, 50, "Tri2")}}};
NewGraph<-display2D( ) Graph;
### III Vocabulary

The \textit{display2D} operator of the relational algebra

#### pr .vocabulary;

<table>
<thead>
<tr>
<th>.attribute</th>
<th>.meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>cart1</td>
</tr>
<tr>
<td>x1</td>
<td>cart1</td>
</tr>
<tr>
<td>x2</td>
<td>cart1</td>
</tr>
<tr>
<td>x3</td>
<td>cart1</td>
</tr>
<tr>
<td>x4</td>
<td>cart1</td>
</tr>
<tr>
<td>y</td>
<td>cart2</td>
</tr>
<tr>
<td>y1</td>
<td>cart2</td>
</tr>
<tr>
<td>y2</td>
<td>cart2</td>
</tr>
<tr>
<td>y3</td>
<td>cart2</td>
</tr>
<tr>
<td>y4</td>
<td>cart2</td>
</tr>
<tr>
<td>sq</td>
<td>sequence</td>
</tr>
<tr>
<td>lc</td>
<td>line_colour</td>
</tr>
<tr>
<td>fc</td>
<td>fill_colour</td>
</tr>
<tr>
<td>tc</td>
<td>text_colour</td>
</tr>
<tr>
<td>fp</td>
<td>fill_pattern</td>
</tr>
<tr>
<td>ls</td>
<td>line_style</td>
</tr>
<tr>
<td>lt</td>
<td>line_thickness</td>
</tr>
<tr>
<td>dl</td>
<td>dash_length</td>
</tr>
<tr>
<td>ft</td>
<td>font</td>
</tr>
<tr>
<td>fs</td>
<td>font_size</td>
</tr>
<tr>
<td>dp</td>
<td>depth</td>
</tr>
<tr>
<td>js</td>
<td>join_style</td>
</tr>
<tr>
<td>cs</td>
<td>cap_style</td>
</tr>
<tr>
<td>fa</td>
<td>forward_arrow</td>
</tr>
<tr>
<td>ba</td>
<td>backward_arrow</td>
</tr>
</tbody>
</table>

Relation .vocabulary has 25 tuples
The \textit{display2D} operator of the relational algebra

III Vocabulary

relation TextVocabulary(.attribute,.meaning) <-
\{(a, cart1), (b, cart2)\};
domain a intg;
domain b intg;
domain textstring strg;
relation Text2(a,b,textstring) <-
\{(5000, 4000, "(5000,4000)"),
    (2000, 4000, "(2000,4000)"),
    (5000, 3000, "(5000,3000)")\};
NewText2 <- display2D(TextVocabulary) Text2;

(Same result as slide “1. Displaying text”)

T. H. Merrett

©07/2
The display2D operator of the relational algebra

IV Updating via the display

E.g., using the two-triangle example we drew earlier:

NewTriangle ← display2D( ) Triangle;
The \textit{display2D} operator of the relational algebra

\section*{IV Updating via the display}

1. Flip the top triangle using the Xfig toolbar;
The \textit{display2D} operator of the relational algebra

IV Updating via the display

2. Draw a third triangle using Xfig tools;
The display2D operator of the relational algebra

IV Updating via the display

3. Edit the bottom triangle, using Xfig tools, to fill with horizontal lines instead of vertical.
The *display2D* operator of the relational algebra

### IV Updating via the display

```plaintext
pr NewTriangle;
+----------------------+----------------------+--------------+
| x1 | y1 | x2 | y2 | x3 | y3 | lc | fc | fp |
+----------------------+----------------------+--------------+
| 5000 | 4000 | 2000 | 4000 | 5000 | 3000 | 1  | 6  | 49 |
| 3000 | 1000 | 1000 | 1000 | 1000 | 2500 | 6  | 1  | 50 |
| 4455 | 2070 | 3417 | 751  | 2793 | 2309 | 0  | 7  | -1 |
+----------------------+----------------------+--------------+
relation NewTriangle has 3 tuples
```

Compare the original version:

```plaintext
pr Triangle;
+----------------------+----------------------+--------------+
| x1 | y1 | x2 | y2 | x3 | y3 | lc | fc | fp |
+----------------------+----------------------+--------------+
| 5000 | 4000 | 2000 | 4000 | 5000 | 3000 | 1  | 6  | 50 |
| 3000 | 1000 | 5000 | 1000 | 5000 | 2500 | 6  | 1  | 50 |
+----------------------+----------------------+--------------+
relation Triangle has 2 tuples
```

©07/2
The *display2D* operator of the relational algebra 20

IV Updating via the display

!! Updates which add new attributes are not allowed (this includes introducing non-default values where defaults were originally used).

!! Updates which need to change a flat relation to a nested relation are not allowed.

!! Updates to nested relations are not implemented.

The *display2D* operator of the relational algebra 21

**V Application: scaling to a box**

\[ x^3 + x^2 + x + 1 \]

\[
\begin{array}{|c|c|}
\hline
f(x & y) \quad box \\
\hline
-2 & -5 & (seq & X & Y) \\
-1 & 0 & 1 & 0 & 0 \\
0 & 1 & 2 & 0 & 440 \\
1 & 4 & 3 & 440 & 440 \\
2 & 17 & 4 & 440 & 0 \\
\hline
\end{array}
\]

let \( xmin \) be red min of \( x \);
let \( Xmin \) be red min of \( X \);
let \( xspan \) be \((\text{red max of } x) - xmin\);
let \( Xspan \) be \((\text{red max of } X) - Xmin\);
let \( xscale \) be \( Xspan/xspan \);
let \( xdispl \) be \( Xmin - xmin*xscale \);
let \( X \) be \( xscale*x + xdispl \);
<< sim. \( Y >>

T. H. Merrett
The \textit{display2D} operator of the relational algebra 22

V Unimplemented: adding sliders

\begin{center}
\begin{tabular}{cccccccc}
\textbf{xslider} & \textit{xloc} & \textit{yloc} & \textit{len} & \textit{rangemin} & \textit{rangemax} & \textit{xzoom} \\
& 0 & 460 & 440 & 1 & 100 & 1 \\
\textbf{yslider} & \textit{xloc} & \textit{yloc} & \textit{len} & \textit{rangemin} & \textit{rangemax} & \textit{yzoom} \\
& 460 & 440 & 440 & 1 & 100 & 1 \\
\end{tabular}
\end{center}

T. H. Merrett
The \textit{display2D} operator of the relational algebra

V Unimplemented: sliders and zooming

\begin{verbatim}
plot
(f box xslider yslider )
(x y) (seq X Y) (.. xzoom) (.. yzoom)
-2 -5 1 0 0 .. 1 .. 1
-1 0 2 0 440
0 1 3 440 440
1 4 4 440 0
2 17

comp post:change:plot/xslider[xzoom]() is
{ let xmin be (red min of x)/[xzoom] in xslider;
  let Xmin be [red min of x] in box;
  let xspan be (red max of x)/([xzoom] in xslider) – xmin;
  let Xspan be ([red max of X] in box) – Xmin;
  let xscale be Xspan/xspan;
  let xdispl be Xmin – xmin*xscale;
  let X be xscale*x + xdispl;
  let X1 be fun succ of X order X;
  << sim. Y >>
  let F be [X,Y,X1,Y1] where X < X1 in f;
  let Box be [X,Y,X1,Y1] in box;

display2D(vocab)[F,Box,xslider,yslider] in plot;
}
\end{verbatim}

T. H. Merrett
The *display2D* operator of the relational algebra 24

V Unimplemented: sliders and zooming

where

\[
\text{vocab} \\
(\text{.attribute} \quad \text{.meaning}) \\
x \quad \text{cart1} \\
y \quad \text{cart2} \\
x\text{slider} \quad \text{slider1} \\
y\text{slider} \quad \text{slider2} \\
X \quad \text{cart1} \\
Y \quad \text{cart2} \\
X1 \quad \text{cart1} \\
Y1 \quad \text{cart2}
\]

We could go on to frame all this in a window with controls for moving and resizing.

T. H. Merrett