

Copyright © 1975, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

GETTING STARTED IN INGRES WITH CUPID

A TUTORIAL

by

Nancy McDonald

Memorandum No. ERL-M546

1 September 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

GETTING STARTED IN INGRES WITH CUPID
A
TUTORIAL

by

NANCY MCDONALD

This document contains an introduction to the data base management system, INGRES, and in particular stresses one of its user language CUPID, the Casual User Pictorial Interface Design. You will be retrieving information from a database via query-diagrams. To "draw" such a diagram, you merely select the appropriate symbols from a 'menu' of pre-drawn symbols ; connect them by either juxtaposition or connector lines; fill in the table and column names through more 'menu' selection; and type in constant value when needed.

This manual is meant to be read while interacting with the INGRES system at a DEC GT42 graphics terminal. It will be assumed that you are sitting in front of an already invoked version of CUPID. If so, a welcoming message should appear on the screen (the invoking procedure will be described in another document).

GRAPHIC TERMINOLOGY

Some graphic terminology is appropriate at this time. The lightpen (abbreviated -LTPEN) is the silver-colored pen-like instrument hanging to the right of the screen. CUPID is designed to necessitate a minimal amount of typing (or keyboard interaction); therefore, most of your actions will utilize the lightpen. The word HIT is used to mean a lightpen selection. There are two types of HITs:

- 1 Touching the item on the screen with the tip of the lightpen. Since the lightpen is light-sensitive, you can select a word, line, or symbol in this way.

- 2 Placing the lightpen on the tracking-cross(8-pointed star-like figure on the screen) and dragging the tracking-cross(t-c) to the place on the screen you wish to select, then depressing the CTRL key and while holding that key down, type an "a"(CTRL-a).

REMEMBER: You can use either method 1 or 2 to select a lighted part of the screen; but, you must use method 2 to select a blank position.

Both of these operations will result in a beep from the machine and the appearance of a small cross very near the hit position which indicates the hit was registered. As soon as you hear the beep, pull the lightpen away from the screen (if you are using the t-c, you may retract the LTPEN as soon as the t-c is in place, even before depressing the CTRL-a).

Please notice the screen configuration. There is an area circumscribed with a large rectangle (the welcome message is inside). Most of your actions will take place within the rectangle, instructions will appear above the rectangle, and control commands will appear to the right of the rectangle. You will be hitting commands also.

LTPEN Exercise

Now try out the lightpen. Touch LTPEN to the screen (gently) where you see the word HIT 1. It will blink if your hit was accurate. If it doesn't blink try again or drag the t-c on top of HIT 1 and type CTRL-a. When it blinks, drag the t-c to a blank portion of the screen (within the rectangle) and depress CTRL-a. Now you should see HIT 2 and you can try to hit it, make it blink, then place a new target elsewhere. When either you or IT have had enough, hit the command HELP.

NOTE: If you happen to make a double hit (your new target will overwrite the blinking one) just proceed as though the old one were not there. But be aware of the possibility of double hits due to the lightpens sensitivity. If this annoys you, turn the brightness (knob at the upper right) down.

HELP

Now you see before you the screen set-up for drawing your queries. You will be selecting names at the bottom left portion of the screen(name-space); you will be selecting symbols(the BOXes and HEXes ,etc.) from the bottom right(symbol-space); and you will be putting them together into a query in the top half of the screen(query-space).

We shall work with only a few of the help commands Hit ALL to see information about all tables in the database ("ndemo" is this

database). When you have finished reading the information, hit RETURN to get back into the help-mode.

Now hit EXAMPLE to see some example queries. Page through several examples with the MORE command.

Note the form of the query in CUPID. Hexagons (containing table names) are abutted to the left of boxes (containing column names). The vertical diamonds with "?" inside are connected by lines to the items (target items) you wish to see displayed; horizontal hexagons are reserved for typing in values. The operators (arithmetic-squares, relational-horizontal diamonds) are straight forward. Aggregation operators (pentagons) are explained in more detail later. Please note that only the HEXes and BOXes are juxtaposed, all other symbols are connected via specific lines (connectors).

When you are finished perusing the examples, hit RETURN to return to the HELP mode.

Once back in the HELP mode, hit CONTINUE to proceed into the query formulation phase.

Query formulation consists of two phases:

- 1) Table Selection
- 2) Query Drawing

TABLE SELECTION

You start the query formulation by selecting the tables you wish to use. Hit "parts" and "supp". The selected table names will appear at the bottom of the screen as you select them. If you get any names other than the two desired (or multiple copies), hit the command REMOVE, then hit the table name to be removed in the selected area at the bottom of the screen (it will be scratched out). You must hit the command REMOVE for each name you wish to delete.

Hit CONTINUE to proceed to Query Drawing.

QUERY DRAWING

The commands that you see on the right side of the screen are:

CONNECTOR: For each connecting line between symbols, hit CONNECTOR, then hit one symbol's attacher point(i.e. the

short line extending from the symbol), then the second symbol's attacher point. If you wish to go around a symbol, or merely draw a "curved" connector, continue making hits as you wish the line to "curve". To draw 2 different connectors(i.e. disconnected lines), you must hit the command CONNECTOR before beginning the second line.

ERASE: This routine will redraw the query (eliminating the bottom half of the screen). It will ask what you wish to erase (Connector, Name, or Symbol); you hit the appropriate word on the right; the screen is redrawn; you hit the item to be erased:

for connectors- hit the first drawn endpoint

for names- hit the lower left point of the first character (NAME refers to any string of characters- i.e. table or column names or constant values)

for symbols- HEX and BOX, hit upper left corner;

all others, hit the center of the symbol; the item hit should blink(keep trying until it does); then hit the command ERASE and the screen will be redrawn without the erased item; either hit RETURN to return to complete the query or hit one of the three types of items to be erased next.

HELP: The HELP command allows one to obtain help while drawing a query.

REJECT: To reject an item (either a name or symbol) after selecting it but before placing it in the query space (after placing you must ERASE), hit REJECT; then the item to be rejected; then proceed.

CONSTANT: To place a constant in a CONS (horizontal hex), hit CONSTANT, then type the value and a carriage return; then place the value inside the CONS-hex; then proceed.

FINISH; Hit FINISH to process query. A new screen configuration will appear in anticipation of the answer to your query. After hitting FINISH either

1. the query will be processed and the answer will appear
2. error messages will tell you the type of pictorial syntax error

you made

3. you may be asked to "define" the constant value used
The three commands you will be hitting are: ALTER to change some aspect(s) of the query just drawn; REDRAW to draw a totally new diagram; and DEFINE to enter the define phase for defining a constant value (to be discussed in another document).

REDRAW: The REDRAW command will clear the query space to allow you to draw a new query.

QUIT: Hit this command to return to the first screen configuration of CUPID. This will allow you to redo the TABLE SELECTION phase or exit entirely.

Before beginning to issue queries there are a few points of procedure.

- 1 HEXagons must be placed in the query space before BOXes which must be next to the HEXes. When placing a BOX next to a HEX or another BOX, point to the upper right corner of the existing item.
- 2 HEXes and BOXes and CONSEs must be on the picture before selecting and placing either names or constantsto be put in them.
- 3 A symbol must be placed in the query space before drawing a connector to it. Connectors must be drawn to and from attacher points. There are from 2 to 4 short lines which jut out of each symbol; the free end of these lines are the attacher points.

Several examples follow.

In this first query, you will be guided in a step by step manner, however, in all other queries, the completed picture will be shown to you and you will be expected to reproduce it in any manner.

Q1: Retrieve the entire parts table.

First select a HEX with the LTPEN by either method described earlier. The HEX will flash. Now bring the t-c to a clear part of the query space and type CTRL-a. The HEX should appear there. (If this doesn't happen, try again -or hit the ERASE command, then hit the RETURN command without erasing --this may help. Electronic problems sometimes make this messy. Do not get frustrated.)

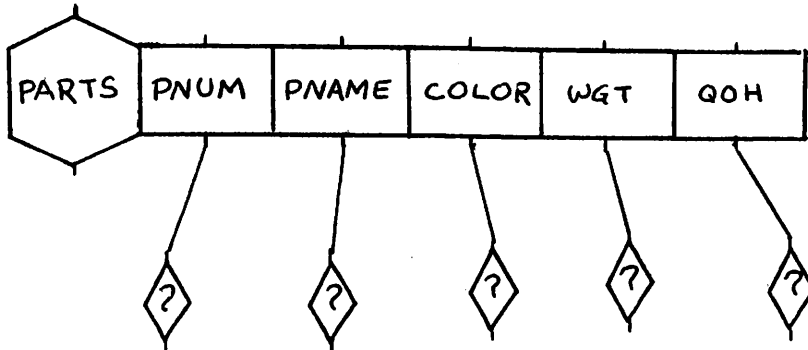
Select a BOX similarly. It will flash; bring the t-c up to the upper right corner of the HEX; type CTRL-a; the BOX should now appear beside the HEX. Repeat this process until you have 5 BOXes attached together and to the right of the single HEX.

Now fill in the table and columns names. Point the LTPEN at the table name- PARTS; it will flash; bring the t-c to point inside the HEX; type CTRL-a; the name-PARTS should then appear inside the HEX. If this doesnot happen try moving it by moving the t-c and typing CTRL-a again. Continue placing all of the column names in BOXes, one to a BOX.

Select the "?"-symbol to indicate which columns you wish to see.

Bring the t-c to a place in the top half of the screen convenient to connect the "?" from the first BOX; type CTRL-a. Proceed selecting and placing until you have 5 "?"'s on the screen

Connect one BOX to one "?" by hitting the command CONNECTOR; hit one endpt of the connector (an attacher point of either the BOX or "?") The line should appear to connect the symbols completely. Repeat this procedure 5 times. Your picture should look like:



Now hit the command FINISH.

The result should be:

pnum	pname	color	wgt	qoh
1	antifreez	pink	10	1
2	wrench	gray	20	32
3	tires	black	685	2
4	ash-tray	black	450	4
5	oil	gray	1	250
6	chamois	yellow	578	3
7	ornament	white	15	95
8	seatcover	blue	19	15
9	race-strp	white	2	350
10	wash-solv	clear	0	143
11	jacks	gray	327	0
12	chrome	gray	427	0
13	tape-play	black	107	0
14	radio	black	147	0

continue

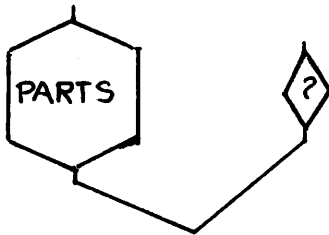
NOTE: The word continue and a beep will always follow the response. Please wait until this occurs before drawing your next query.

Notice that the "parts" table has information about the components in a hypothetical auto parts supply store. Each row of this table (or tuple in this relation) contains information on a given part including its part number (pnum), its part name (pname), its color, its weight (wgt), and the quantity that are on hand (qoh).

Now you are ready to draw the next query. Hit ALTER to correct, change or add to the last query drawn; hit REDRAW to draw a picture from scratch; hit QUIT to get to the first screen configuration (this will allow you to proceed from the beginning and step through the TABLE SELECTION phase again). The DEFINE command presently acts like the QUIT command.

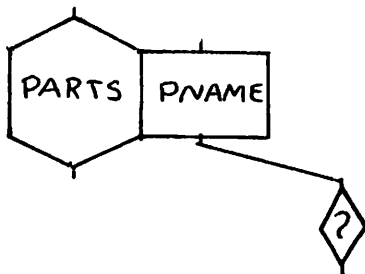
Hit REDRAW.

Q1.5 A short-hand diagram for that query is:



Using a CUPID picture we will be able to obtain portions of this table which are of interest to us. (There is almost no limit on how large the tables can be; these examples are done on small ones so that this tutorial does not become too large. In fact, the actual limit on the size of a table is approximately 30,000,000 bytes for those who are interested.)

Q2: Retrieve PNAMEs out of the PARTS table.



```

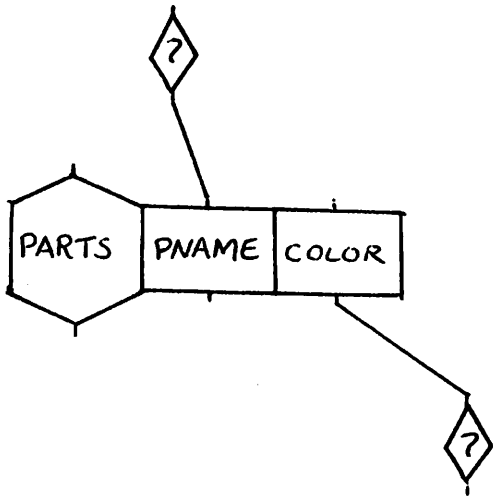
pname
-----
antifreez
wrench
tires
ash-tray
oil
chamois
ornament
seatcover
race-strp
wash-solv
jacks
chrome
tape-play
radio
continue

```

Everything has worked correctly and we have obtained the column of the parts table which contains the names of the parts.

We can retrieve more than one column at once(as in Q1) by simply indicating a sequence of boxes containing column names with attached "?" diamonds. At this point you may either hit the command ALTER or REDRAW.

Q3: Thus, to obtain part names and colors we draw:

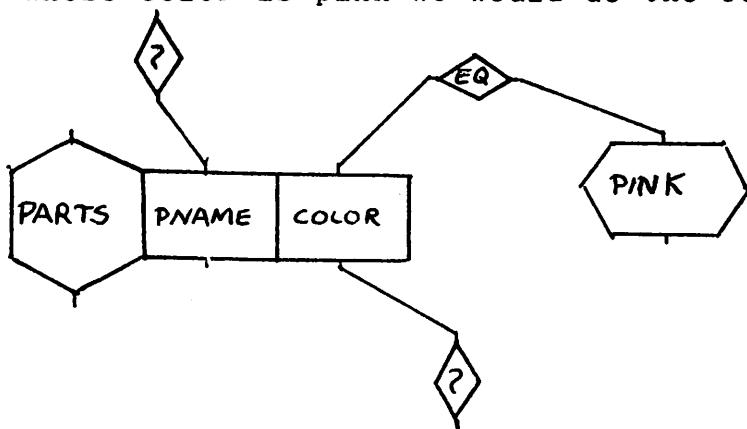


pname	color
antifreez	pink
wrench	gray
tires	black
ash-tray	black
oil	gray
chamois	yellow
ornament	white
seatcover	blue
race-strp	white
wash-solv	clear
jacks	gray
chrome	gray
tape-play	black
radio	black

continue

So far we have produced interactions which give us columns of the "parts" table. We now indicate how to obtain only portions of columns. The basic mechanism is to connect appropriate limiting operators and any constant values to the BOXes targeted with "?"-symbols.

Q4: If we wanted the previous query only performed for those parts whose color is pink we would do the following:



An explanation of how to place the word - pink - is in the following NOTE on the next page.

pname	color
antifreez	pink

continue

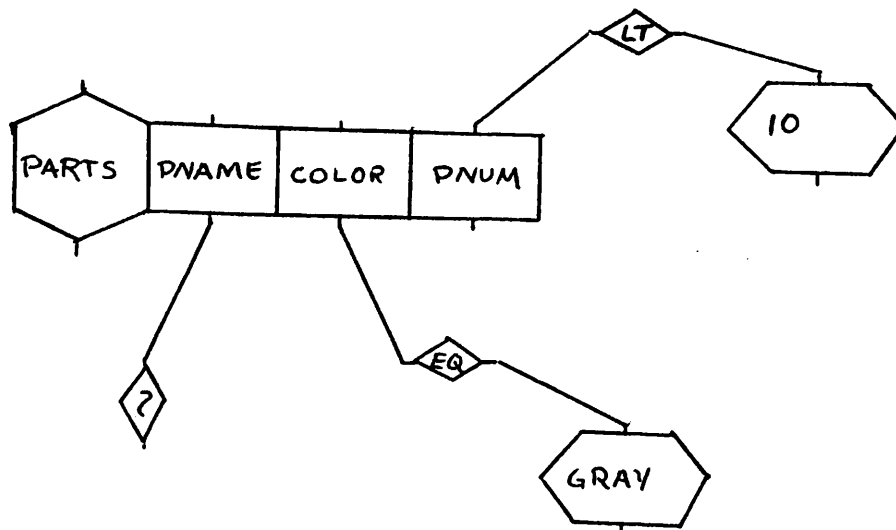
These limiting operators reduce the number of rows which are examined to only those which satisfy the qualification given i.e. to those which satisfy the part of the query not singled out with "?" -or:



Only antifreez has this property so it is the only entry in the output.

NOTE: To place the word 'pink' in the CONS-box; hit the command CONSTANT; type - pink - and a 'cr'(carriage return); now place the t-c inside the CONS and depress CTRL-a. The first letter of every constant value must be entirely in the symbol.

Q5: Obtain the pnames of parts which are gray and whose pnum is less than 10.



```

|pname
|-----|
|wrench
|oil
continue

```

Three points should be carefully noted about the above interaction:

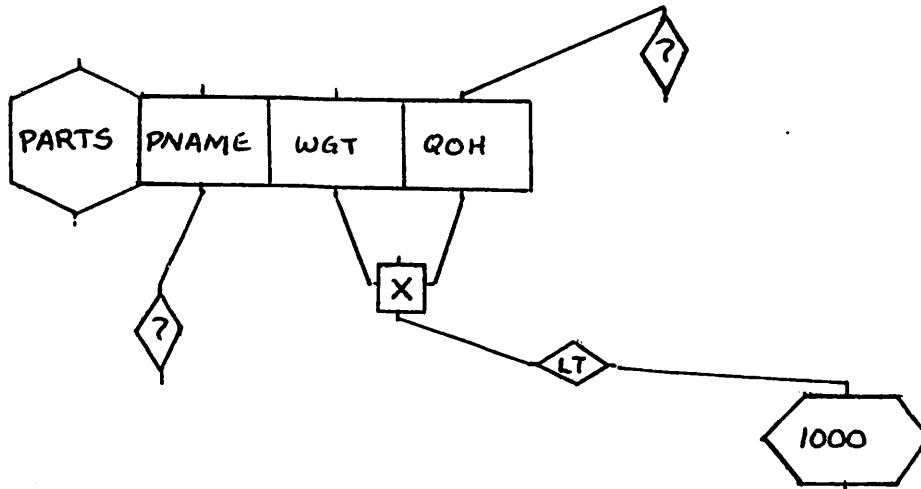
1) Note the relational operator `LT` in the above interaction. Valid relational operators include:

- ~~EQ~~ (equals to)
- ~~LT~~ (less than)
- ~~GT~~ (greater than)
- ~~LT~~ ~~EQ~~ (less than or equal to-not implemented)
- ~~GT~~ ~~EQ~~ (greater than or equal to-not implemented)
- ~~NE~~ (not equal to)

2) There is no limit to the complexity of the expressions which can be constructed using relational and boolean expressions, theoretically since the present implementation is restricted.

3) All of the relational operators except ~~EQ~~ and ~~NE~~ are ordered operators. This means that whatever is connected on the left is the first operand and the item connected on the right is the second.

Q6: We now do an example concerning arithmetic operators in CUPID. This example finds pname and qoh of parts whose total weight (wgt times qoh) is less than 1000.



pname	qoh
antifreez	1
wrench	32
oil	250
seatcover	15
race-strp	350
wash-solv	143
jacks	0
chrome	0
tape-play	0
radio	0
continue	

It should be noted that arithmetic operators can be used in the target list ("?" portion of an interaction as well as in the qualification portion

Note also that any operations (even a "?"-symbol) you may wish done to the result of an arithmetic operation is 'hung' off of the operator [as in the picture part meaning less than 1000].

Valid arithmetic operators include:

- (addition)
- (subtraction)

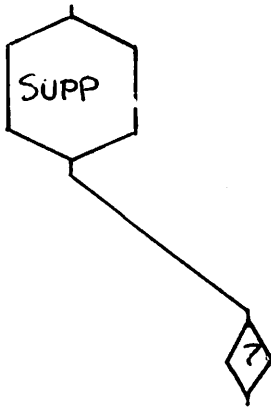
$\boxed{*}$ (multiplication)

$\boxed{/}$ (division)

$\boxed{**}$ (exponentiation)--not implemented

We turn now to interactions which involve more than one table at a time. It is in these interactions that CUPID is especially useful because of its ability to connect information in different tables.

Q7: First we print a second table that will be used in the sequel.



supp relation

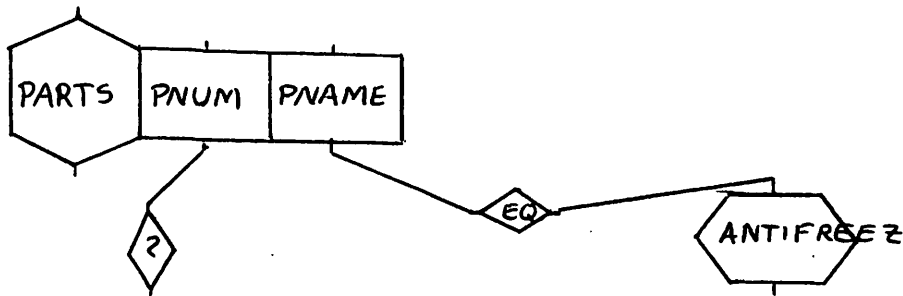
snum	pnum	jnum	shdt	quan
8	1	1003	74-12-31	1
8	1	1004	75-01-15	1
8	1	1007	76-02-01	1
8	2	1003	74-12-29	128
8	2	1004	75-01-15	256
8	2	1007	76-02-01	1024
8	6	1003	74-12-25	2
8	6	1004	75-02-01	4
8	8	1004	74-12-20	5
8	9	1004	74-12-31	500
8	11	1004	75-01-01	2
8	11	1007	76-02-01	3
8	12	1004	75-04-31	3
8	12	1007	76-02-01	2
9	5	1004	75-02-05	400
20	5	1001	75-01-10	20
20	5	1002	75-01-10	75
41	5	1003	75-01-02	50
62	3	1002	74-06-18	3
67	4	1005	75-07-01	1
67	5	1005	75-07-31	20
122	7	1003	75-02-01	144
122	7	1004	75-02-01	48
122	9	1004	75-02-01	144
131	8	1001	75-03-15	2
131	8	1002	75-03-15	1
131	8	1004	74-11-22	4
131	9	1001	75-04-31	200
131	9	1002	75-03-31	100
440	6	1001	74-10-10	2
475	1	1001	73-12-31	1
475	1	1002	74-07-01	1
475	2	1001	73-12-31	32
475	2	1002	74-05-31	32
475	3	1001	73-12-31	2
475	4	1002	74-05-31	1

continue

This table gives information on conditions under which the hypothetical auto parts store can buy more parts. It indicates the supplier number (snum) from whom each part (pnum) is available, the quantity (quan) in which it can be ordered, the date (shdt) such an order could be shipped and the job number (jnum) to which such an order could be charged. Notice that the column pnum appears in both the parts table and this table. Using this

information we can "connect" the two tables. For example, we might want to know the supplier numbers of suppliers who sell antifreeze.

Q8: One way to proceed is to interrogate the parts table to find the part number of antifreeze as follows:



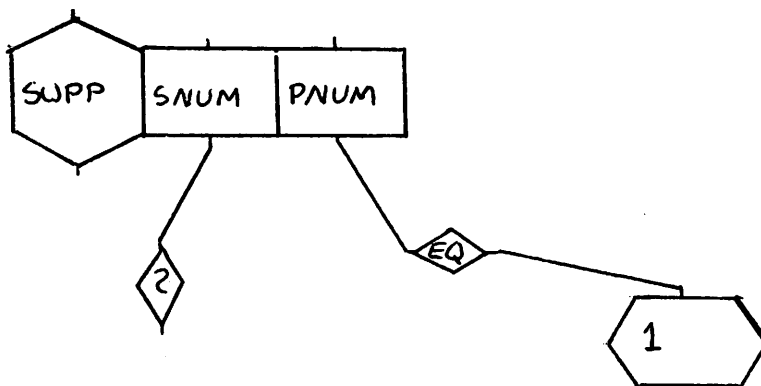
The answer returned is:

pnum
1

continue

Hence, part number 1 is antifreeze.

Q9: Then we could interrogate the supply table seeking the suppliers of part number 1 as follows:



snum
8
8
8
475
475

continue

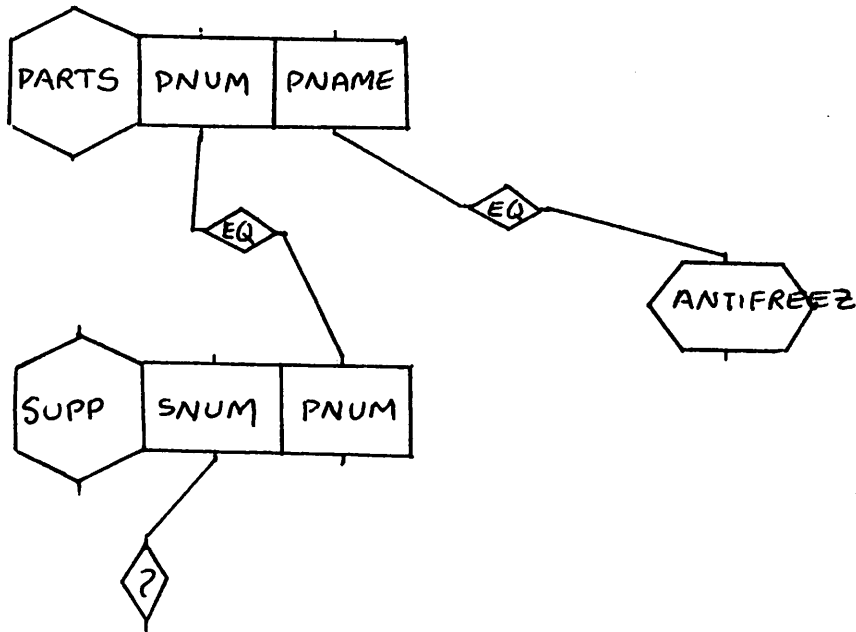
Notice that suppliers 8, 475 supply antifreeze.

Notice also that suppliers 8 and 475 are repeated more than once. Because of the internal way that INGRES is organized, much faster response time can be supported if the "answer" is printed on the terminal with duplicate values sometimes present. In this case, the user must look at the response and note the duplications. On the other hand, should the user wish the system to detect and delete the duplicates, the user need only retrieve his answer into a temporary relation and then print that relation. At this time the appropriate protocol for CUPID has not been implemented. We must live with duplicates.

It is rather inconvenient to have to issue two query formulations to get the information we require.

What is even more inconvenient is the necessity of obtaining the first output, namely the number 1, and then manually substituting this into the second query. It would have been extremely inconvenient if the antifreeze had had several part numbers; we would have had to substitute them all.

Q10: The following indicates one way around this inconvenience.



Resulting in the following:

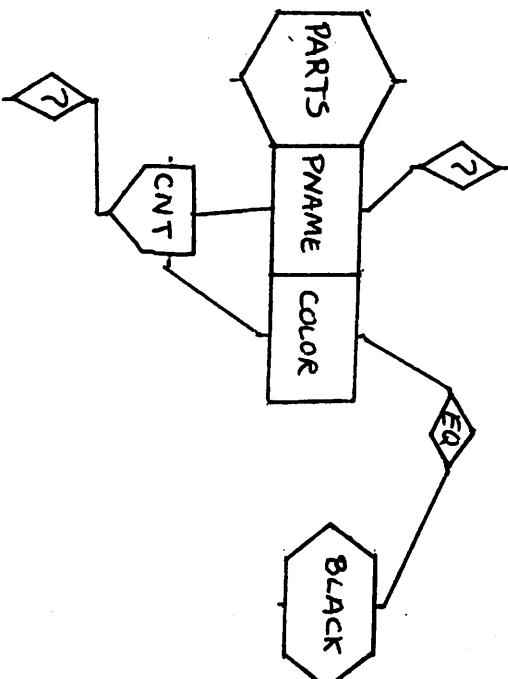
```

snum
-----
      8
      8
      8
     475
     475
continue
  
```

So far in this document we have considered how to retrieve portions of a table (or tables) that are of interest. The examples have indicated the power of CUPID for retrieval purposes. The only feature which has not yet been considered is aggregation.

We now illustrate the use of this construct in two examples.

Q11: The following command finds the number of part names from the parts table which are black.



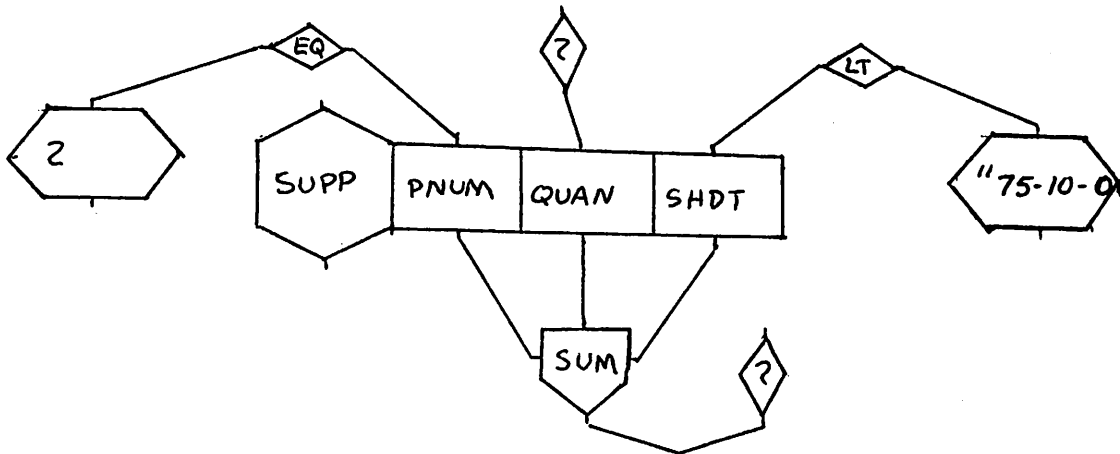
```

c
-----
4
  
```

continue

You should note that there are two "?"-symbols and only one column of result. This is due to information necessary to perform aggregation. Each aggregate operator needs a targeted ("?"-symbol attached) BOX. The aggregate may also need some untargeted #BOXes (known as qualification). Notice the line connecting the aggregate operator CNT to the box containing COLOR. This indicates that the qualification of color = black' modifies the aggregate and not the whole query(if the connector where not there). Without that connecting line, we would have obtained the count of PNAMEs for each time the color of an item is black or the result would have shown 14 listed 4 times. You might like to try this query also. If so, just erase the connector and reissue the query.

Q12: The next command finds the sum of quantities of part number 2 to be supplied before October 1, 1975.



NOTE: Due to the special type of information in column SHDT (dates) you must type "75-10-01 with the first character being " .


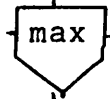
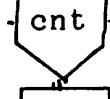
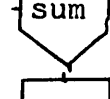
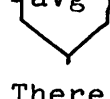
```

s
-----
  448
continue

```

The following points should be noted about aggregates:

a) aggregate operators include:

- | | |
|---|--------------------|
|  | minimum |
|  | maximum |
|  | count |
|  | sum |
|  | average(sum/count) |

b) There are illegal aggregations. For example, avg is only

allowed for quantities which are numeric. An attempt to find the average of a quantity that is alphanumeric (for example pname) will result in an error.

- c) An aggregate can appear anywhere in a CUPID interaction.
- d) Aggregate operators may be (1) attached at one end only - meaning any qualification does not refer to the aggregate; or (2) attached to boxes at several points - meaning the box not marked with a "?" starts the qualification referring to the aggregate only.
- e) The resulting column heading is a CUPID generated heading using letters of the aggregate operator's name.

Research sponsored by an IBM Fellowship and the National Science Foundation Grant DCR75-03839.