Database Design Appendix C, app3.tex

This file is ©1977 and 1983 by McGraw-Hill and ©1986, 2001 by Gio Wiederhold.

This page is intentionally left blank.

Appendix C

Symbols Used

Mathematical Symbols Used

$\lceil\ \rceil$	CEIL, next higher integer
$\lfloor\ \rfloor$	FLOOR, next lower integer
$\approx$	approximately equal
¿¿	much greater than
!	factorial
#	number of
$\log_y x$	logarithm base y of x
$\log x$	natural logarithm of x, base $e = 2.71828182846$
$\sum_k f(k)$	sum of all $f(k)$ for the integer k's specified
$\ominus$	one of the comparison operators $> \geq = \neq \leq <$
$\wedge$	and, true if both sides are true
$\vee$	or, true if either side is true
$\vert$	where, precedes a conditional clause
$\bigcap$	set intersection
$\bigcup$	set union
$-$	set difference
$\times$	cartesian product
$\subseteq$	select tuples from a relation
Π	project attributes from a relation
$\underset{a=b}{\bowtie}$	join two relations based on equality of the attributes a, b
≻——	reference connection
——∗	ownership connection
——∋	subset connection
$\subset$	subset of
$\in$	member of
$\forall$	for all
$\Rightarrow$	becomes
$\{\ \ \}$	enclose a set
[ ]	enclose a reference

Programming and Syntax Symbols As Used

In general we follow the convention of PL/1, a language originally developed by IBM to serve both scientific and commercial programming tasks. Some examples use Ada. a language sponsored by the US Defense department, COBOL, a widely used commercial language, and Pascal, a popular language for teaching.

a + b	addition
a - b	subtraction
a * b	multiplication
a / b	division
MOD(a,b)	modulo, integer remainder of division
a ** b	exponentiation, a to the power b
a = b	depending on context in PL/1, assignment or equality comparison
a > b	greater than comparison, true if a greater than b
a $\geq$ b	greater or equal comparison, true if a greater or equal to b
a $\wedge$ b	and, true if both a, b true (& in PL/1)
a $\vee$ b	or, true if either a, b true (| in PL/1)
$\neg$ c	not, true if c false and vice versa (≠ in Ada, Pl/1)
s || w	catenation, string w appended to string s (& in Ada)
s | c	where, do s if the predicate (conditional) clause c is true
R $\bigcup$ S	union of relations R and S
R $\bigcap$ S	intersection of relations R and S
R $\times$ S	cross product of relations R and S
R — S	difference, remove tuples matching S from R
$\underline{\mathcal{S}}$ R.ex	select tuples of R according to expression ex
$\prod$ R.a	projection of attributes a of R
R.a⋈S.b	Join R and S, on equality of attribute values in a and b
R.a⟗S.b	Outerjoin including all tuples
,	field separator
:$\rangle$	key and goal fields separator
;	statement separator
.	termination of computational section
ss, ...	section ss may be repeated
[ss]	section ss is optional
{ss/tt}	sections ss, tt are alternatives
::=	is defined by
/* Note */	explanatory comments
a.b	qualification of variable b by a higher-level variable a,i.e., employee.name
"Word"	character string constant
_	(underline) pseudo-alphabetic character without syntactic meaning used for legibility within variable names. (In COBOL – is used for this function.)

Variables Used in Performance Formulas

A	average space required for attribute name	Sec. 3-1-3
a	number of different attributes in a file	Sec. 3-1-1
a'	average number of attributes in a record	Sec. 3-1-1, 3-6-3
B	blocksize	Sec. 2-2
b	blockcount	Sec. 2-2-2
btt	block transfer time $= B/t$	Eq. 2-13
Bfr	blocking factor $\approx B/R$	Eqs. 2-5, 2-6, 2-7, 2-20
C	Cost factors	Sec. 5-4-6, 5-5-2
c	computational overhead per record, when not negligible	Sec. 2-3-4
D	space required for data	Eq. 5-1, Sec 5-3-3
d	number of records that have been invalidated	Sec. 3-1-3
$\textsc{f}$	subscript denoting a fetch for a specific record	Sec. 3-0-2
G	space required for an interblock gap	Sec. 2-2-3
h	classification variable	Sec. 5-4-3
$\textsc{i}$	subscript denoting insertion of a record	Sec. 3-0-2
j	number of cylinders	Sec. 2-2-1
K	projection list	Sec. 7-3-2
K	Kilo or thousand (1024) times	
k	number of tracks per cylinder	Sec. 2-2-5, Table 2-1
L	load frequency factors; selection list	Sec. 5-1; 7-3-3
M	multiprogramming factor	Eq. 5-19
M	Mega or million (1 048 576) times	
m	number of available slots for records	Sec. 3-5-1
$\textsc{n}$	subscript denoting getting the next serial record	Sec. 3-0-2
n	number of records in a file	Sec. 3-1-3
o	number of records that overflow	Secs. 3-1-3, 3-2-3, 3-3-3, 3-5-3
P	space required for a pointer	Sec. 2-3-3
p	collision cost, also probability	Eqs. 3-73, 3-74, 3-79, Fig. 3-23
q	production demand by a file application	Eq. 5-2, 5-4 to 5-6
R	space required for a complete record; relation	Sec. 3-0-2; 7-1-1
$\textsc{rw}$	subscript indicating rewriting	Sec. 2-3-6
r	rotational latency time	Eq. 2-3
SI	storage space for index	Eq. 3-52
s	average seek time	Eq. 2-2
s'	effective seek time	Eqs. 2-15, 2-16
T	the time required for various operations; set of tuples	Sec. 3-0-2; 7-3
T_{sort}	the time required to sort a file	Eq. 3-11
t	transfer rate from a storage unit to processing memory	Sec. 2-2-5
t'	bulk transfer rate	Eqs. 2-17, 2-18, 2-19
$\textsc{u}$	subscript denoting an update of a record	Sec. 3-0-2
u	utilization	Eqs. 5-3, 5-7 to 5-18, 5-20
uf	utilization factor	Eqs. 6-25, 6-29
V	average space for value part of an attribute	Sec. 3-1-3
v	number of records updated	Sec. 3-2-3
w	wait time in queues	Eq. 6-26
W	wasted space due to gaps per record	Eqs. 2-9, 2-10, 2-11, 2-21
$\textsc{x}$	subscript denoting an exhaustive search	Sec. 3-0-2
x	number of levels in an index structure, master level	Eq. 3-27, 3-49, 3-97
$\textsc{y}$	subscript denoting a reorganization of a file	Sec. 3-0-2
y	fanout ratio	Eq. 3-26, 3-48