

## TECHNICAL QUESTIONS

### 1. COBOL

#### COMMON BUSINESS ORIENTED LANGUAGE

**STATIC LINK**      **STATIC IS INCLUDED IN THE GLOBAL LINK OF THE PROGRAM/**  
 Subprogram linked into main program/always resident  
**DYNAMIC**            **SEPARATELY LINKED**  
                          **STATICALLY LINKED PROGRAMS ARE CALLED WITH THE PROGRAM NAME IN QUOTES**  
                          Subprogram loaded into memory only when called by calling program

**FIGURATIVE CONSTANT**      **COBOL RESERVED WORD HAVING A PRE-ASSIGNED VALUE ZERO**

**GO TO VS. PERFORM**            **GO TO IS ONE WAY WHILE PERFORM BRINGS YOU BACK TO THE NEXT STATEMENT TO BE PERFORMED AFTER THE PERFORM**

**COBOL INTERNAL SORT**      **SORT FILE NAME ON ASCENDING KEY - ALLOWS YOU TO INCLUDE LOGIC TO BE PERFORMED BEFORE AND AFTER SORT/ I.E. ON A CALCULATED FIELD**

**SORT FILE NAME 1 GIVE FILE NAME 2**

Set up an internal sort

In JCL allocate sort work space, in data division code SD, in procedure division use SORT statement with the on ascending/descending key, using / input procedure

Why use an internal sort? To alter data immediately before sorting it/reduce need for Separate programs

#### READ A FILE WRITE A RECORD

**CLASS TEST**            **CHECKS FOR SPECIFIC DATA**      -      **ALPHABETIC/NUMERIC**

**USAGE IS**            **USAGE IS DISPLAY/ZONED DECIMAL**  
**USAGE IS COMP-3/PACKED**  
                          2 digits per byte sign in last half byte  
                          # digits/2 rounded up  
                          Positive right half byte is C  
                          Negative right half byte is D  
**USAGE IS COMP/BINARY**  
                          Left most bit indicates sign 0 is positive 1 is negative

#### MOVE STATEMENTS

	FROM	TO	
	X(10)	x(7)	left justified
	X(7)	x(10)	truncate last three digits
9(10)	9(5)		truncate first/right five digits
	S99v99	s9v99	truncate 1 <sup>st</sup> and last digits – retain sign
	S9v99	s999v999 009.990	
105	PIC9(4)		F0 F1 F0 F5
105	PIC9(4) COMP-3		00 10 5C
19	PIC9(4) COMP		00 13 HEX
length of s9(5)v99 comp-3		= 7 + 1 = 8/2 = 4	

length of s9(7)v9(3) comp-3 = 10 +  
length of s9(04)v99 = 6

implied decimal point place where decimal point is implied, either from the rightmost position or  
the v indicator in the picture clause

**BUFFER** AREA IN STORAGE WHERE BLOCKS ARE BUILT BEFORE WRITTEN OUT

Table handling two dimensional tables lower level occurs within occurrence of first higher level

Initialize a table spaces to A or X fields  
Zeroes to numeric fields  
Initialize a table through redefines clause in data division; reading values from a file; using  
VALUE clause on the entry, which names the table

01 TABLE-ONE VALUE '1234'  
05 TABLE-TWO OCCURS 4 TIMES PIC X

Use OCCURS clause in the Data Division ex. 05 table occurs 1 to 10 times  
Depending on fielda ascending key is fieldb  
Indexed by index pic x

Can have three levels in cobol vs, 7 in cobol ii

Files match/merge if keys not equal, lowest key first  
If keys equal, primary key first

Carriage control positioning, advancing, or placing carriage control character in first byte  
RECFM=FBA in JCL

Special registers used in cobol  
Day, date, time, current date, sort return, sort core size

#### ARITHMETIC FUNCTIONS

ADD/SUBTRACT/MULTIPLY/DIVIDE

#### FILE ACCESS

SEQUENTIAL/RANDOM/INDEXED

#### IDENTIFICATION DIVISION

(NAMES PROGRAM/AUTHOR/DATE)  
PROGRAM-ID  
PARAGRAPHS  
ENTRIES  
CLAUSES

#### ENVIRONMENT DIVISION

(CONNECTS TO JCL/CONFIGURATION)  
FILE-CONTROL  
USES SELECT/ASSIGN CLAUSES TO CONNECT FILES TO JCL  
BLOCK=0 TO ALLOW JCL SPECIFICATIONS

#### DATA DIVISION

(DESCRIBES DATA ELEMENTS/FILES)

#### FILE-SECTION

FILE DEFINITIONS (FD)  
WORKING-STORAGE

DEFINING RECORDS HERE MAKES IT EASIER TO  
FIND RECORDS IN DUMPS

DATA STRUCTURES LEVEL 01  
RENAMES LEVEL 66  
CONDITIONAL TEST LEVEL 88  
CONSTANTS

**LINKAGE-SECTION**

DESCRIBING DATA IN A CALLED PROGRAM THAT  
WAS PASSED FROM A CALLING PROGRAM/DBMS (CICS, IMS)/JCL  
PARM

THE CALLED PROGRAM CONTAINS THE LINKAGE SECTION  
EXIT VIA GOBACK OR EXIT PROGRAM

CALLING PROGRAM WILL USE CALL USING  
Parm on JCL EXEC statement

Define parm field in the linkage section with first two bytes defined as PIC  
S9(4) comp for the parm length; code USING option on the procedure division header

**WORKING STORAGE****DATA DESCRIPTIONS****FD FILE DESCRIPTION**

Label records are standard

Record contains

Block contains

If 0, the blocksize is picked up from the JCL

Dynamic -- records in a file can be accessed seq or randomly

Variable length recording mode v/multiple 01 or occurs depending on

Write depending on a field name defining the length/

Write from record description

**SD SORT/MERGE FILE DESCRIPTION****RD REPORT DESCRIPTION****ELEMENTARY ITEMS CANNOT BE SUB-DIVIDED AND HAVE PICTURES****Open statements**

Input, output i/o, extend,

Use a rewrite statement with i/o file will be read but also will be written back  
onto the file at the same disk locations that they were read from

Extend - like mod

**PROCEDURE DIVISION**

(EXECUTABLE STATEMENTS)

ELEMENTARY/ACTION VERBS

OPEN

READ FILES

WRITE RECORDS

CLOSE

**LOGIC:**

AND/OR/NOT

IF/ELSE

CONTINUE NO EXECUTABLE SENTENCE IS PRESENT/OUT OF IF CLAUSE

NEXT SENTENCE GOES TO STATEMENT FOLLOWING THE NEXT PERIOD

SET TO TRUE

PERFORM

LOOP CONTROL

PROCEDURE

THRU N TIMES

UNTIL

PERFORM VARYING FROM/UNTIL

UNTIL PRE-TEST

IF TRUE OUT/IF FALSE PROCESS

Counter is updated from FROM value

Test is done at the beginning

INLINE PERFORM

PERFORM VERB CAN BE USED WITHOUT NAMING A PARAGRAPH

PERFORM UNTIL CNT < 1

END-PERFORM

Use inline perform when you do not want to branch out of a paragraph to

Perform an iterative process

case nested ifs

EVALUATES

CONTROL:

GO TO

EXIT

STOP RUN

ends cobol programs

GOBACK

exit a called program

CONTINUE

NEXT

**DATA MANIPULATION:**

MOVE TO  
 MOVE CORRESPONDING    STRUCTURE MOVE  
 INITIALIZE  
 ACCEPT  
 DISPLAY  
 HEX NOTATION  
 REDEFINES

**File Status Return Codes**

00        last i/o operation successful    10 end of file        30-90 errors

**ARITHMETIC:**

ADD/SUBTRACT/MULTIPLY/DIVIDE        add x1, x2 to x3  
 COMPUTE                                compute x3 = x1 + x2  
 ROUNDED  
 SIZE ERROR CONDITION

**TABLES:**

INITIALIZE THROUGH REDEFINES IN DATA DIVISION/READING VALUES  
 FROM FILE/USING VALUE CLAUSE ON ENTRY THAT NAMES TABLE  
 SET UP A TABLE USING THE OCCURS CLAUSE IN THE DATA DIVISION  
 INDEX BY

**EXAMPLE:**

ROLL UP TABLE USING READ INTO IN A PERFORM UNTIL EOF  
 IF NOT EOF, MOVE READ FILE FIELD TO ARRAY FIELD  
 IN SEARCH PARAGRAPH, SET INDEX FIELD TO 1  
 SEARCH TABLE VARYING INDEX WHEN FIELD = ARRAY(SUB)

SET        used to set value in index

**SEARCH**

SERIAL SEARCH

BINARY SEARCH

SUBSCRIPTS    OCCURRENCE IN TABLE

INDEX        DISPLACEMENT OF AN ITEM IN TABLE/MORE EFFICIENT

SET/SEARCH/SEARCH ALL/PERFORM VARYING CHANGE VALUE  
 OF INDEX

SEARCH        SEQUENTIAL SEARCH

SEARCH ALL    BINARY SEARCH

**SUBROUTINES:**

DYNAMIC CALL/CANCEL

CALL        causes control to be passed from one object program to another

PARM

**DEBUGGING:**

READY/RESET TRACE

DISPLAY

EXHBIT

ON

RETURN-CODE    MOVE CONDITION CODE <= 4095

**DATA TYPES:**

DISPLAY DATA

PICTURE

Z        NUMERIC SUPPRESS

X        ANY TYPE OF DATA

A        CHARACTER/SPACE

9        NUMERIC

V        IMPLIED DECIMAL

S        SIGN

ALSO .,\$/+-cr db

**USAGE**

PACKED DECIMAL COMP-3 105        00105C

C        POSITIVE        RIGHT 1/2 BYTE

D        NEGATIVE

BINARY INTEGER COMP        0015

HIGH ORDER 0        POSITIVE

HIGH ORDER 1        NEGATIVE

INDEX        INDEX

ZONED DECIMAL DISPLAY F0F1F0F5

ALIGNMENT

**CHARACTER MANIPULATION:**

INSPECT (REPLACE, TALLYING)	DATA VALIDATION
STRING/UNSTRING	CONCATENATION or separation
TRANSFORM	CHARACTER REPLACEMENT/ alter characters from one set of values to another
CONTINUE	NULL BRANCH
NEXT SENTENCE	PASSES CONTROL TO NEXT SENTENCE
Set to true	used to set a conditional item to true, i.e. 88 eof statements
I/O:	
BUFFERS	I/O AREA GROUPED STORAGE
MOVE	
LOCATE	
END OF FILE	
COBOL SORT	
IN DATA DIVISION CODE A SD	
IN PROCEDURE DIVISION USE ON ASCENDING/DESCENDING KEY USING	
SET CONDITION CODE IN JCL	MOVE SPECIFIED VALUE TO RETURN-CODE
DIFFERENCE BETWEEN COBOL II AND COBOL VS	
DIVISIONS/SECTIONS/CLAUSES ARE NOW OPTIONAL/OBSOLETE	
LABEL RECORDS/FILLER OPTIONAL	
RESERVED WORD TO ALLOWED IN ADD/GIVING STATEMENT	
>=/<= ALLOWED	
VALUE CLAUSES ALLOWED ON OCCURS CLAUSES	
IMPROVED INTERFACES WITH CICS AND SORT/MERGE/	
COBOL DOCUMENTATION	
USE MEANINGFUL DATA/PARAGRAPH NAMES	
USE CHANGE LOG AT TOP OF PROGRAM	
USE NOTES IN COLUMNS 73-80	
HOUSE STANDARDS	
WORKING STORAGE SECTION	CONSTANTS ONLY
USE SPACE/ZERO NOT " " "0"	
SIGN NUMERIC FIELDS	
USE () IF PICTURE EXCEEDS 3	
PROCEDURE DIVISION	
TOP DOWN/STRUCTURED PROGRAMMING	
ONE ENTRY AND EXIT FOR EACH SECTION OF CODE	
NO GO TO UNLESS XFER TRANSFERRED TO EXIT WITHIN PARAGRAPH	
DON'T USER ALTER	
ON STATEMENT (VERB) PER LINE	
AVOID COMPLEX NESTED IFS	
SPELL GREATER THAN/EQUAL, ETC.	
NO LITERALS/COMMAS	

## PL/1

### COBOL/PL1 SIMILARITIES

DATA STRUCTURES	ESP. RECORDS
RECORD-ORIENTED I/O	
PICTURE DECLARATIONS	
REPORT-GENERATING FACILITIES	

COBOL	PL/1	DB2	BAL	VB
PIC S9(4) BINARY COMP-4	FIXED BIN(15,0)	SMALLINT		
PIC S9(9) BINARY	FIXED BIN(31,0)	INTEGER		
PIC S9(N) PACKED-DEC COMP-3	FIXED DEC(N)	DEICMAL(N)	P/PACK/UNPK	
PIC S9(N)V(M) PACKED-D L	FIXED DEC(N+M,M)			
PIC X(N)	CHAR(N)	CHAR(N)	C	
POINTER	POINTER			
ACCEPT XX FROM DATE	DATE	DATE		
ACCEPT XX FROM TIME	TIME	TIME/TIMESTAMP		
EVALUATE	SELECT			SELECT

WHEN WHEN OTHER END-EVALUATE.	WHEN OTHERWISE END		CASE
<b>1,0,-+ CARRIAGE CONTROL CHARACTERS</b>			
VALUE	INIT		
OPEN/CLOSE	OPEN/CLOSE		OPEN/CLOSE
READ/INTO/AT END WRITE/FROM	READ/INTO WRITE/FROM		GET PUTPR
GO TO	GO TO/LEAVE		B/BNE/BE
STRING			
SUBSTR	(M:N)		
TRANSFORM	TRANSLATE		
GOBACK	RETURN		
MOVE/TO	=		MVC/MVN
IF/ELSE	IF THEN/ELSE		CLC IF/THEN ELSEIF IIF
PERFORM/UNTIL	DO UNTIL		
PERFORM/VARYING	DO WHILE		DO/ DO
=><EQUAL LESS THAN GREATER THAN	=><		WHILE UNTIL WHILE WEND
AND OR NOT ()	& ^ ()		
			FOR
ACCEPT/FROM d/t	DATE/TIME BUILTIN		NEXT
DISPLAY	PUT DATA		
REDEFINES	DEF/BASED(ADDR) OVERLAY		
ROUNDED	ROUND		
SUBTRACT/ADD/MU DIVIDE/- + * /	ADD/DIVIDE/MULTIPLY - + * /	BUILTINS	ZAP/AP/MP
OCCURS X TIMES	(x)		DIM/
CALL/USING	CALL/fetch		REDIM
			STATIC
<b>DATA ATTRIBUTES:</b>			
FIXED DECIMAL	PACKED	1101 BIT PATTERN NEGATIVE/1100 POS	
FIXED BINARY	2 OR 4 BYTES	FIXED BIN	
FLOAT DECIMAL	0.1275E3		
ZONED DECIMAL			
HEXADEMICAL	0-9 A-F		
FLOAT BINARY			
BIT	EIGHT BITS TO A BYTE	'1'b	
CHARACTER	EBCDIC CHARACTER SET		
CHARACTER PACKED NO SIGN	NO POSITION RESERVED FOR SIGN		
CHARACTER BINARY			

## LANGUAGE

>< &  
hierarchy of operations

## I/O:

STREAM  
LIST DIRECTED GET/PUT LIST  
EDIT-DIRECTED GET EDIT/PUT EDIT  
DATA-DIRECTED GET/PUT DATA  
TRANSMISSION  
RECORD-ORIENTED (COBOL)  
CTLASA/CTLW60 CONTROL CHARACTER  
I/O STATEMENTS  
READ/WRITE/REWRITE/DELETE  
SEQUENTIAL INPUT/OUTPUT  
READ FILE INTO  
WRITE FILE FROM  
SEQUENTIAL/DIRECT UPDATE  
READ FILE INTO KEY  
REWRITE FILE FROM

## EXPRESSIONS:

ARITHMETIC  
STRING  
BOOLEAN  
BOOL  
LOGICAL OPERATORS

DATA CONVERSION

## DATA ORGANIZATION:

RECORDS (COBOL)  
ARRAYS  
ASSIGNMENTS  
SCALAR TO ARRAY  
ARRAY TO ARRAY  
BY NAME  
LIKE  
EXPRESSIONS

STRUCTURES

## EDITING:

PICTURE (COBOL)  
ASSIGNMENT

## PROGRAM ORGANIZATION:

PROCEDURES OPTIONS  
DO LOOP(LIKE COBOL PERFORM/VARYING)  
DO/BY  
DO WHILE  
DO UNTIL

BLOCK STRUCTURE  
STRUCTURED STATEMENTS  
RECURSION

CASE SELECT WHEN OTHERWISE  
table handling

ARRAYS  
SUBSCRIPTS  
SUBSCRIPT EXPRESSIONS  
ARRAY EXPRESSIONS

## STORAGE ALLOCATION:

STATIC INITIALIZED ONCE  
AUTOMATIC  
BASED  
DEFINED  
POINTER READ FILE(X) SET(B)

## SUBROUTINES:

INTERNAL  
CALL  
EXTERNAL  
DCL XXX ENTRY EXTERNAL  
FETCH XXX  
CALL XXX  
RELEASE XXX

## FUNCTIONS:

## USER-DEFINED

## BUILTINS

DATE YYMMDD

TIME HHMMSSSTTT

SUBSTR

ROUND

mod mod (29,6) = 5 6 x 4 + 5

INDEX STARTING LOCATION

VERIFY RETURN BIN ZERO IF FOUND

ADDR

SUM

LENGTH

REPEAT

TRANSLATE SUBSTITUTES A VALUE

CHAR CONVERT TO CHARACTER

## REPORT GENERATING (COBOL)

## ERROR HANDLING:

## ON CONDITIONS

KEY

RECORD

ENDFILE

SIZE

CONVERSION

ZERODIVIDE

FIXEDOVERFLOW

SUBSCRIPTRANGE

## DEBUGGING

CHECK PROGRAM TRACE

PUT STATEMENTS

INTEREST/EXPERIDITOR

## INSERTION CHARACTERS

, . BLANK

## COMPILERS:

CHECKOUT

OPTIMIZING

## ALGORITHMS

## BINARY TREE SEARCH

ARRANGE A TABLE IN ASCENDING ORDER/

DIVIDE IN TWO/ COMPARE SEARCH ARGUMENT

WITH TABLE ARGUMENT UNTIL AN EQUAL COMPARES OCCURS

## MATCHING

## KEYED MATCH

ON KEY

READ FILE INTO KEY

## INDEXED MATCH

READ FILE A

READ FILE B

IF FILE A &lt; B WRITE A/READ A

IF FILE A &gt; B READ B

ELSE MATCH/READ A/READ B

## PARSING ROUTINES

REPORT MODULE/LEVEL BREAKING

TABLING UP/LOADING

CONVERSIONS

PACKED NO SIGN TO CHARACTER

## 2. CICS

### Map Field

I		
L	BIN(15)	if 0, field is missing on input
F	X(1)	Attribute byte
	Intensity	
	Field protection	protected/unprotected, auto skip
	Shift	alphanumeric/numeric

### Extended attributes

Control over cursor position	-1 to FIELDL PIC S9(4) USAGE COMP
Special display characteristics-	blinking, color
Hex c1 ebcidic a MDT =y unprotected	

Hex c9 ebcidi I unprotect, high intensity ,and MDT y

Like an OS that maintains its own region or partion

**MDT** attribute of a field, which will always transmit the contents of the field from the physical map to the logical map when, turned on

Turned on when data is keyed into the field by the terminal operator

Data sent to CPU by 3270 only from field with MDT on

**Queues-** author/non-fina/sales/returns/payments.merch

**Command level translator**

**RRN** relative record number

**PROGRAM**

HANDLE CONDITION  
 DEFINE TIOA AREA 1920 BYTES  
 ADDRESS TWA/CWS  
 RECEIVE INTO TIOA  
 CHECK TIOA OR EIBTRNID  
 VALIDATE INPUT  
 EXEC CICS LOAD  
 WRITE TO QUEUE  
 WRITEQ TD OR REWRITE

Customer Information Control System

**QUASI-REENTRENT** programs share the same procedure division but have separate working storage

**PSEUDO-CONVERSATIONAL** program is not kept in main storage while waiting for input from terminal/program is transaction driven /release resources between executions of a task information passed between programs and task via commarea or temporary storage

**CUSTOMER INFORMATION CONTROL SYSTEM**

At automatic task initiator start up a data-driven transaction

IBM Link error resolution

Dynamic Transaction Backout-- backouy in-flight task that terminates abnormally

**DPL Command** specifies a remote region on the sysid option/ client/cocs region ships request to remote server region exec cics start transid from/interval

Transid to link to remote program in another region

Set up tran id in pct to remote region

Exec cics link pgm/sysiid/commarea/transid

**MRO** multiple region operation cics to cics commandallows cics substem to use resources of each other function shipping/transaction routing/distributed transaction processing

Intercommunication Facility ship data via CEMT ITD

Tioa contains data being display on the terminal

**POP UP WINDWS** OHF Software from Software Diversified Company, Minn

Field specific pup-up windows/sticky cursors to cut and paste

Multi-level menus

Send page/send text

Console messages writeq td queue('cnsl') from(' ')

CICS Abend-Aid  
 CA-Interrest?Speditor  
 Verify  
 Model Region Testing  
 Applied Vision Center  
 LPAR

EXEC CICS command(paramenets  
 END-EXEC

IBM TELEPROCESSING MONITOR

TABLE CHANGES

PCT	PROGRAM CONTROL TABLE TRANSACTIONS One entry for every type	
PPT	PROGRAM PROCESSING TABLE PROGRAMS/MAP SETS Use count	
JCT	JOURNAL CONTROL TABLE JOURNALS	
SNT	SIGN ON TABLE	
SIT	SYSTEM INSTALLATION TABLE	
FCT	FILE CONTROL TABLE FILES	
TCT	TERMINAL CONTROL TABLE TERMINALS/SYSTEM CONNECTIONS	
DCT	DESTINATION CONTROL TABLE TYPE=INTRA/REMOTE/INDIRECT/EXTRA DETIID TRIGLEV TRANSID Td defined here	SET BY TRIGGERS
	INTRAPARTITION DATA GENERATED BY ONE TASK AND PROCESSED BY ANOTHER IN THE SAME PARTITION OR REGION	
	EXTRAPARTITION GENERATED IN ONE PARTITION AND PROCESSED ANOTHER	
	INDIRECT (ALIAS) DESTINATIONS	
	PSEUDO-CONVERSATIONAL PROGRAM NOT IN MAIN STORAGE WHILE WAITING FOR TERMINAL INPUT	
	HANDLE CONDITIONS (BRANCHES TO ERROR RTN)	in procedure
division	can also branch to another program i.e. eexec cics handle program Executes specific logic for specified conditions	
	DUPREC	
	NOSPACE	
	NOTOPEN	
	NOTFND	
*	PGMIDERR	
*	MAPFAIL	occurs on receive map when there are no fields
on the screen for bms to map to	ae19 abend eibfn 18	
*	ERROR	* =MPs
	FILENOTFOUND	
	IOERR	unable to do a read or write
	DISABLED	
	dsiderr	
	ILLOGIC	try to read from or write into an empty vsam ksds file
need at least 1 record		
	INVREQ	
	BAD FCT/KEYLENGTH/READ COMMAND	
	LENGERR	

IGNORE/NOHANDLE        to turn off handle  
 CHECK EXCEPTION IMMEDIATELY AFTER COMMAND EXEC  
 PUSH    SUSPEND HANDLE  
 POP     REINSTATE HANDLE  
 EX.     PUSH  
           HANDL  
           LINK  
           POP  
 AID     ATTENTION IDENTIFIER keys  
 PA1-3   attention keys  
 PF1-24   function keys  
 ANYEY

i.e. exec cics handle aid pa1(print)  
 anyley(error)

printc                    cics issuer print

alternative to handle aid is to use EIBAID field  
 EIBAID = DFHPA!

#### PROGRAM TYPES

MESSAGE PROCESSORS	TERMINAL COMMUNICATION	audit
	trail/update records	
UPDATES	PROCESSES TRANSIENT DATA	
UTILITIES	I/O HANDLERS	
Subroutines and i/os		
	Code for both twa and commarea	
	If eibcalen = 0 use twa	
	If twaptr nullabend	

#### ERROR MODULES

Dynamic criteria processing

#### TERMINAL CONTROL

##### IBM SERVICE TRANSACTIONS

CECI	COMMAND INTERPRETOR—UPDATE VSAM FILES OR DELETE TD QUEUE RECORDS
CEBR	TEMPORARY STORAGE BROWSE FACILITY
CESN/CSSN	SIGN ON
Cebt	master terminal
CSSF	SIGN OFF
CSMT/CEMT	NEW COPY            use PH phase in Start up auxiliary trace atr,On
CSTT	SYSTEM STATS
CRTE sysid	ROUTING/SHIPPING TO DIFFERENT REGION
CEDF	EXECUTION DIAGNOSTIC FACILITY/DEBUGGING
CMSG	message switching
CWTO	message to console
CEDA	DYNAMIC ADDITION OF MAP SITES    RDO RESOURCE DEFINITION define trans/prog/mapsets/term.sess

NATIVE MODE    from ttoa rather than read from map

RECEIVE

SEND

Erase  
 Freekb  
 cursor

CONVERSE

#### PROGRAM CONTROL

##### TRANSFER CONTROL VIA COMMAREA

Via twa only when going to macro-lvelp

NAME OF AREA TO BE PASSED AND/OR AREA RESULTS ARE TO BE  
 RETURNED

Storage areas outside program are coded in linkage section

LINK SUB-MODULES INVOKED/SUBROUTINES AT SAME LOGICAL LEVEL VIA TWA/BOTH PGMS REMAIN IN MAIN STORAGE  
LINK TO ANOTHER PROGRAM ANTICIPATING RETURN  
Return control to initiating program i.e. 1/o modules

CHECK EIBCALEN

TWA TRANSACTION WORK AREA TO PASS DATA BETWEEN PROGRAMS  
Entry in the PCT table

UIB User interface block

CICS + DL1

COPY DLIUID

Schedule PSB	CALL CBLTDLI	COBOL
	CALL PLITDLI	PLI

DLI SERVICE CALLS

- ISRT
- LOAD
- REPL
- DLET
- Shed
- Term
- Gu/gn/gnp

XCTL CONTROL PASSED TO ANOTHER PROGRAM at the same logical level without any return mechanism (i.e. RETURN)

Program/commarea/length  
Conditions: notauth, pgmiderr  
go back to CICS tables

RETURN  
LOAD  
RELEASE  
COMMAREA COMMUNICATION AREA DFHCOMMAREA IN EXECUTE INTERFACE BLOCK LINKAGE SECTION

EIBCALEN LENGTH OF COMMAREA  
RETURN CODE  
EIBRCODE  
EIBDATE YYDDD

CWA COMMON WORK AREA WORK AREA USED TO SHARE SYSTEM-WIDE DATA AMONG TASKS  
CONSOLE/SYSTEM DATES ASSIGN  
Predefined and reserved for system updates

ADDRESS CSW/CWA/TCTUA/TWA  
LOAD bring load module in ppt into main storage  
RELEASE  
ABEND

FILE CONTROL

READ/UPDATE

FILE  
INTO/SET DATA AREA/PTR REF  
LENGTH  
RIDFLD  
KEYLENGTH  
SYSID  
EQUAL/UPDATE

REWRITE - UPDATE  
READNEXT  
READPREV  
RESETBR reset after nofound condition followed by STARTBR

command

**DELETE**  
**UNLOCK** - **RELEASE EXCLUSIVE CONTROL**/if a record was retrieved for update purposes but not processed, it releases a record to other transactions  
**WRITE**  
**STARTBR**  
**BEGINBR** = startbr + read next  
**ENDBR**  
**MAPPING SUPPORT**  
**BMS** **BASIC MAPPING SUPPORT**  
**TERMINAL APPLICATION INTERFACE**  
**ASSEMBLER PROGRAM MAPSETS**  
**SDF** **SCREEN DEFINITION FACILITY**  
Msl map spec library  
Dct devicte charact table  
Equates characters  
**RECEIVE MAP**  
**SEND MAP/PARTN/TEXT/CONTROL/PAGE**  
**SEND TEXT** **WRITE FORMATTED DATA TO SCREEN**  
**WITHOUT USING A MAP**  
  
**MAPONLY** send only the physical map including the map's defaults  
**DATAONLY** send only the logical map data  
  
**PURGE MESSAGE**  
**TRANSIENT DATA** **INTRAPARTION DATA QUEUE**  
Cannot modify an item/can only add new items to bottom of queue  
Must be read sequentially  
Uses trigger in dct/intra  
**WRITEQ TD**  
**READQ TD**  
**DELETEQ TD**  
**TEMPORARY STORAGE** **SCRATCH PAD/PREPRINTED FORMS/intermediate results/terminal**  
paging/mesdsage routing/data collection i.e. updt cics comes down  
**WRITEQ TS**  
**MAIN/QUEUE/ITEM/LENGTH/FROM**  
**READQ TS**  
**DELETEQ TS**  
**ITEM** number of records in temporary storage  
Relative record number  
½ word binary  
**ITEMERR** is EOF indicator  
  
**AUX** storage queue is stored in remote storage device  
**MAIN** stored in main storage  
**STORAGE CONTROL**  
**GETMAIN** **STORAGE**  
**FREEMAIN**  
**TASK CONTROL**  
**SUSPEND** give up control so a higher priority task can executed  
**ENQ** **TASK WANTS A PARTICULAR RESOURCE EXCLUSIVE**  
Enqueue protects a resource from concurrent use by more than one task  
**DEQ** dequeue frees the resource from the enqueue  
**INTERVAL CONTROL**  
**ASKTIME** **REQUEST TIME OF DAY**  
**DELAY**  
**POST** indicates when times has expired  
**WAIT EVENT**  
**START** start a task on a local or remote system  
**RETRIEVE**  
**CANCEL**  
**TRACE CONTROL**  
**ENTER**  
**TRACE**

BIF DEEDIT - EDIT BUILTIN IN FUNCTION remove non-num right j  
 ustifty  
 DUMP CONTROL  
 DUMP MAIN STORAGE OR TABLES  
 SYNC POINT CONTROL  
 SYNCPOINT ROLLBACK  
 DEBUGGING  
 TRACE  
 DUMP  
 EXEC CICS ABEND ABCODE('AAA')  
 PRINT A SCREEN VIA 3270 PRINT REQUEST/  
 EXEC CICS ISSUE PRINT  
 VERIFY SCREEN CAPTURE  
  
 USE HANDLE CONDITIONS TO FLAG ERRORS  
 IF NOT OK THEN EXEC CICS HANDLE CONDITION ERROR(E1)  
 USE EXEC CICS ASKTIME AND STEP VIA CEDF OR EXPEDITOR  
 IN CHECKING A CICS DUMP, DTERMINE WHICH FILES WERE SKIPPED/  
 UPDATED/SECTION WERE EXECUTED  
 SUBTRACT ADDRESS OF PROGRAM FROM PSW AT EBEND  
 TO GET OFFSET OF NEXT EXECUTABLE INSTRUCTION  
 TRACE TABLE  
 AUTOMATIC/STATIC STORAGE AREAS  
 TERMINAL STORAGE DISPLAYS  
 EIB FIELDS EXEC INTERFACE BLOCK  
 EIBFN last command executed  
 EIBTIME  
 EIBDATE  
 EIBTRNID  
 EIBRCODE RESPONSE CODE  
 EIBRESPS DFHRESP(NOTFND/duprec/notauth))  
 Fullword binary fib (31)  
 CSA COMMON SYSTEM AREA  
 CWA COMMON WORK AREA  
 TCA TASK CONTROL AREA  
 TWA TRANSACTION WORK AREA  
 FWA FILE WORK AREA  
  
 RECOVERY PROCESSING  
 SYNCPOINT  
 JOURNAL  
 WAIT JOURNAL  
 QUASI-REENTRANT SHARE PROCEDURE DIVISION BUT HAVE  
 DIFFERENT WORKING STORAGE  
 ATTRIBUTE BYTE  
 INTENSITY  
 FIELD PROTECTION  
 SHIFT  
 EXTENDED ATTRIBUTE CURSOR LOCATION  
 PROTECTION/ALPHA/NUMERIC/HIGH INTENSITY/  
 MODIFIED DATA TAG/AUTO SKIP  
  
 COBOL CODE NOT ALLOWED IN CICS READ/WRITE/OPEN/CLOSE  
 SYNCPOI NT COMMAND DIVIDES A CICS TASK INTO LUWS LOGICAL UNITS OF  
  
 ISSUE SYNCPOINT WHEN PERFORMING MULTIPLE LOGICAL TASKS  
 ENQUEUE/DEQUEUE ENQ COMMAND ISSUED WHEN RESOURCE TO BE  
 PROTECTED  
  
 SCREEN ENTRIES  
 MAPFAIL CONDITION  
  
 ERASE  
  
 HANDLE CONDITIONS

WORK

CICS APPLICATION FILE CONTROL FACILITY  
 CONTROL OF ONLINE RESOURCES FILES/TRANSACTIONS/PROGRAMS  
 OPEN/CLOSE ENABLE/DISABLE  
 TABLE-DRIVEN

CICS ABENDS

ABM0 MAP NOT LOCATED  
 AE NO HANDLE CONDITIONS AEIA-AEY3

DB2 ERRORS

+100 ROW NOT FOUND  
 -818 PRECOMPILER TIMESTAMP DIFFERENT FROM BIND TIMESTAMP

MVS/QUICK REF  
 IBM/BOOKMANAGER

CICS/MCS/COMPILER UPGRADES  
 ACF2

ACCESS CONTROL FACILITY

DEFAULT-BASED MECHANISM FOR CONTROLLING USER ACCESS  
 OF SYSTEM RESOURCES

PRIVILEGE/OWNERSHIP TABLES

USER ID STRING PROFILES

ABEND-AID/DEBUGGING

CAUSES/LOCATION OF ABEND/ABENDING PGM/LIST DATA AREAS/  
 PREVIOUS AND CURRENT RECORDS

FOR NON ABEND-AID TO FIND THE INSTRUCTION THAT ABEND PGM:

SUBTRACT EPA (ENTRY POINT ADDRESS) FROM PSW (PROGRAM  
 STATUS WORD). LOCATE SOURCE STATEMENT USING LINKAGE  
 EDITOR AND TRANSLATOR MAPS (DMPA/PMAP). THE PREVIOUS  
 INSTRUCTION FROM DISPLACEMENT ADDRESS IS THE ONE THAT  
 ABENDED.

001	I/O	
B37	SPACE	
806	PGM NOT FOUND	
322	CPU TIME OUT	
222	OPERATOR CANCE;	
OC7	DATA EXCEPTION	NON-NUMERIC
OC4	PROTECTION EXCEPTION	INVALID SUBSCRIPT
OC1	OPERATION EXCEPTION	ACCESS UNOPENED FILE

NOT CATLG 2 FILE CREATED BUT NOT CATELOGED

STROBE

AUTOMATED SYSTEM PERFORMANCE ANALYSIS

PROGRAMART/MASS

**PINPOINT CAUSES OF EXCESSIVE TIME AND RESOURCE UTILIZATION/REDUCE APPLICATION RESOURCE DEMAND**

**TAKES SNAPSHOTS OF RESOURCE UTILIZATION WHILE APPLICATIONS ARE EXECUTING AND PRODUCES PERFORMANCE PROFILE REPORTS TO PINPOINT AREAS OF CODE WHERE DEMAND IS CONCENTRATED**

**ELIMINATING EFFICIENCIES**

**TMON**

**THE MONITOR FOR CICS/LANDMARK**

**NAVIGRAPH      PC-BASED WITH GUI INTERFACE FOR ANALYSIS OF SYSTEM PERFORMANCE**  
**NAVIGATE      PROVIDES SMART LINKS BETWEEN TMONDS/JUMPRING FROM ONE ENVIRONMENT TO ANOTHER IN A SEAMLESS FASHION**

**POWERFUL STORAGE DISPLAYS/GRAPHICS**  
**USER FRIENDLY/GOOD ONLINE HELP FACILITIES**

**TOR      terminal owning region**

**RDO      resource definition online**

**define resources via CEDA PCT/PPT entires**

### 3. JCL

#### JOB CONTROL LANGUAGE

JCL IS AN IBM LANGUAGE TO DESCRIBE A BATCH JOB AND  
AND ITS REQUIREMENTS TO THE OPERATING SYSTEM

#### OVERRIDES

Condition code indicates a result of its execution, a program can issue a return code using CON  
Bypass or execute a step

Symbolic parameters allows for dynamic modification of values at execution time  
Name preceded by ampersand / allow procedures to be modified  
i.e. loadlibs, sysout classes  
override a symbolic param after the EXEC in the proc, symbolic = 'value'  
override to PARM parameters in EXEC PARM.stepname='value'  
MODIFY JCL STATEMENTS IN CATALOGED OR IN STREAM PROCS  
PARM = &DECK&CODE  
DSN=UNSERLIB.&LEVEL  
If the symbolic parm is not used, JCL errors

CONTROL-M col= %%VOLSER)

JCL IF/THEN/ELSE/ENDIF allows users to conditionally execute steps within JCL

BLOCK SIZE MAX FOR TAPE 32760  
DISK 27920

JEM/JOBS CAN/JCLCHECK/TYPRUN=SCAN JOB/SCAN MACROS

DATA PACKER DATA COMPRESSION

BATCH PIPES/MVS IBM

HOOKS INTO I/O INTERFACE  
CACHES INTO VIRTUAL STORAGE  
PROCESSES BEFORE FILES CLOSE  
JCL CHANGE SUBSYS ON IEBGENER

#### STANDARDS

FORMS/POLICIES/PROCEDURES

BY PASS LABEL (BLP) NL/SL  
SL IBM STANDARD LABEL  
NL NON STANDARD LABEL DON'T NEED ACF2

GDGS GENERATION DATA GROUP SUCCESSIVE, HISTORICALLY-RELATED  
CHRONOLOGICAL DATA SETS UP TO 255  
CURRENT 0  
DSCB CONTROL BLOCK BUILD A GDG BASE ENTRY  
CREATE A MODEL DATA SET LABEL VIA IEHPROGM/IDCAM5  
AMS DEFINED VIA DEFINE  
TO CHANGE LIMIT USER ALTER/NOEMPTY/NOSCRATCH

SMS IBM SYSTEM MANAGED STORAGE  
(MVS/DFP DATA FACILITY PRODUCT)

SYSTEM MANAGED DATASETS/SPACE MANAGEMENT

DATA COMPRESSION

DATA PACKER

HRECOVER/HRECALL

## SORT

SUM FIELD=NONE NO DUPES  
 BI UNSIGNED BINARY  
 CH UNSIGNED CHARACTER  
 PD SIGNED PACKED DECIMAL  
 FOR VARIABLE ADD 4 TO SORT FIELDS

## SYNCSORT

REPLACES MVS SORT UTILITY DF/SORT  
 CREATE HEADERS/TRAILERS

## CONFIGURATION MANAGEMENT

## TUNING/SORT SPACE CALCULATION/DASD CAPACITY ANALYSIS

## SYSTEM EXIT (IEFACTRT)

RETURN CODE  
 EXCPS I/OS  
 I/O DEVICE CONNECT TIME  
 TASK CONTROL (TCB) CPU TIME  
 SRB (SYSTEM SERVICES) CPU TIME  
 CLOCK (WALL) TIME  
 PAGING OPERATIONS  
 SWAPS

## JOB COLLECTION OF JOB STEPS

RESTART  
 MSGCLASS SYSTEM MESSAGES  
 CLASS  
 NOTIFY  
 TYPRUN=SCAN  
 REGION  
 TIME  
 ALSO: RACF PROTECTION

## EXEC

PARM  
 COND  
 ONLY IF PRIOR TERMINATES  
 EVEN IF PRIOR TERMINATES execute step even if preceding step has failed  
 (0,NE,stepname)  
 COND=(#,GT/LT/NE/GE,STEP#)

## DD

## DATA DEFINITION

DSN  
 &&TEMP temp pds &&dsn(mem1) temporary dataset  
 \*.DDNAME  
 DUMMY test program flow without processing data  
 NULLFILE defines a dummy data set

DISP  
 UN/CATLG MOD SHR OLD NEW PASS DELETE  
 Status,normal terminal disp, abnormal termination disp  
 (MOD,DELETE) delete disk that may or may not exist

SPACE  
 RLSE

DCB  
 DEN=3 1600 BPI  
 DEN=4 6250 BPI  
 RECFM F/FB/S/V/VB/U  
 LRECL  
 BLKSIZE SYSTEM DETERMINED FOR DISK I.E. DO NOT NEED DCB OR =0  
 TRTCH=COMP RECORDING TECHNIQUE NOCOMP IDRC  
 improved data recording capability

UNIT  
 DEFER DON'T MOUNT UNTIL NEEDED  
 AFF UNIT AFFINITY DEVICE REQUIRED FOR DD UNIT=ADD=DD1  
 REQUEST THAT A VOLUME BE MOUNTED USING THE SAME DEVICE

LABEL  
 RETPD  
 EXPDT  
 (# of files, NSL/BLP/SL,EXPDT=98000)Tape SLH F1 T h2 f1 = label=2,blp

VOL  
 RETAIN RETAIN VOLUME  
 (,,10)

TAPE

INSTREAM PROC/PEND SAME SET OF JCL CATALOGED IN A PDS PROCEDURE LIBRARY

Aaa PROC MAX # IS 15  
 PEND  
 EXEC proc=aaa  
 DD \* OVERRIDES to include instream data

PROCEDURE nested procedures MVS/ESA v. 4 procedure that calls other procedures up to 15 levels

RESTART

BACKWARD REFERENCE REF A VOLUME FROM A PREVIOUS VOL=REF

REFERBACK //OUT1 OUTPUT CLASS=A  
 //RPT1 DD SYSOUT=(,),OUTPUT=(\*.OUT1)

JES2 CONTROL the input and output processing of jobs  
 JOB ENTRY SUBSTEM  
 /\*ROUTE PRINT RMT1  
 /\*JOBPARM PROCLIB=PROC01

JOBLIB/STEPLIB

## VSAM VIRTUAL STORAGE ACCESS METHOD

**DBT Advisor to tune vsam definitions ci size and freespace requirements**

**DataPacker/VSAM –BMC DASD savings up to 80%**

AMP PARM

AMP = (BUFNI (# I/O BUFFERS), bufnd # 1/0S TO BE PROCESSED)

DELETE

DEFINE CLUSTER  
 NAME  
 INDEXED  
 KEYS  
 FREESPACE FREE BYTES (X,Y) X = % OF fb IN ci/y= %ci/ca  
 RECORDSIZE  
 SHAREOPTIONS (3 3)  
 SPEED  
 NOIMBED  
 UNIQUE  
 NOERASE  
 NOWRITECHECK  
 NOREPLICATE  
 NONSPANNED  
 SHARE

DATA  
 NAME  
 CONTROLINTERVALSIZE(4096)  
 CYLINDERS  
 OWNER  
 VOLUMES(\* \* \*)

INDEX SAME AS DATA/NO OWNER  
 SET MAXCC = 0 AT LOAD,COND=(0,ne)

ALTERNATE INDEX

DEFINE AIX  
 NAME  
 FREESPACE  
 KEYS  
 OWNER  
 RECORDSIZE  
 RELATE  
 SHAREOPTIONS (2 3)  
 UNPGRADE  
 NUNQK

DATA  
 NAME  
 CISZ  
 VOLUMES

INDEX  
 NAME/CYL/VOLUMES

BLDINDEX INDATASET/OUTDATASET  
 DEFINE PATH NAME/PATHENTRY

DEFINE A RELATIONSHIP BETWEEN AN ALTERNATE INDEX AND ITS BASE CLUSTER

LISTCAT

AMS ACCESS METHOD SERVICES IDCAMS UTILITY  
 DEFINE/REPRO/PRINT/LISTCAT

VIRTUAL STORAGE ACCESS METHOD  
 ORGANIZATIONS:

ESDS	ENTRY SEQUENCED DATA SET	LOADED IN SEQ
	LOGS/JOURNALS OR rba	RELATIVE BLOCK ACCESS
KSDS	KEY SEQUENCED DATA SET	INDEXED
	MOST	
RRDS	RELATIVE RECORD DATA SET	RECORD LOACATION
		RELATIVE TO
		START OF DATASET
	RRN	RELATIVE RECORD NUMBER

FILE MAINTENANCE, DATA PORTABILITY, DATA SECURITY

ALTERNATE INDEX EXAMPLE  
 ACCOUNT/TELEPHONE

DELETE  
 DEFINE ALTERNATEINDEX  
 DEFINE PATH/PATHENTRY  
 BLDINDEX

IMPROVE KSDS/VSAM  
 PLACE INDEX AND DATA ON A SEPARATE PACK  
 USE IMBED OPTION TO STORE SEQUENCE SET RECORD FOR EACH CA  
 ASSIGNMENT OF FREE SPACE  
 STORAGE DATE COMPONENTS ON MULTIPLE VOLUMES

USE dbt ADVISOR TO TUNE vsam FILE DEFINTIONS, I.E. cisize AND freeSPACE

UsE STROBE TO ANALYZE CPU USAGE

Vsam ASSIST FOR REORGS

VERIFY IF A PROGRAM ABENDS IN MIDDLE OF PROCESSING FILE AY REMAIN OPEN

IDCAMS/CLUSTER  
 RECORDS/VOLUME/KEY  
 SHAREOPTION  
 CROSS SYSTEM/REGION SHARING

CI CONTROL INTERVAL CONTINUOUS AREAS OF AUXILIARY STORAGE  
 WHERE RECORDS RESIDE

CA CONTROL AREA

VSAM I/O PLUS VIOCTL DD CARD ADDED TO CALCULATE BUFFER AREAS AND  
 TRANSMITTING DATA AND INDEX CONTROL INTERVALS BETWEEN VRITUAL AND AUX  
 STORAGE

## CLIST

COMMAND LIST FILE WITH LIST OF COMMANDS TO  
 COMPUTER THROUGH CONSOLE

CLIST COMMAND LIST

INTERPRETED RATHER THAN COMPILED

REXX RESTRUCTURED EXTENDED EXECUTOR

IBM LANGAUGE ON MVS/TSO

SUPPORTS STRUCTURED PROGRAMMING INSTRUCTIONS  
 AND OPERATIONS AND PERFORMS LIKE CLIST

ISPF TABLE --- ISPF PANEL

|

v

OBJECT CREATION FORMAT DDL DATA DEFINITION LANGUAGE  
 DB2

WORKBENCH TOOL ACCESS

ACCESS USER DBMS

PLANS/TIME STAMP VERIFICATION

CONVERT JULIAN TO GEORGIAN

## IBM UTILITIES

Use Platinnum products Fast Load and Quick Copy rather than standard IBM utilities

IDCAMS  
 REPRO/INFILE/OUTFILE/REPLACE/FROMKEY/TOKEY  
 PRINT/FKEY/COUNT  
 VERIFY TO ENSURE FILE PROPERLY CLOSED

IEB DATASET UTILITIES  
 IEH SYSTEM UTILITIES

IDCAMS DEFINES, CONSTRUCTS, BACKUP, PRINTS, DELETES

lefbr14

IEBCOMPR COMPARES LOGICAL RECORDS

IEBCOPY COPY, COMPRESS, MERGE  
 IEHLIST LIST ENTRIES IN SYSTEM CATALOG/TOC  
 IEBGENER  
 IEBUPDT

COPY INDD

COPY

ISPF/PDF OPTION 3.2  
 ISPF COPY  
 IDCAMS  
 IEBCOPY

IEHMOVE MOVES OR COPIES DATA SETS OR VOLUMES

IEHPROGM/IDCAMS GDGS

IEBGENER COPY/REBLOCK DS

EXEC  
 SYSPRINT  
 SYSUT1  
 SYSUT2  
 SYSIN DD DUMMY

## **TSO/ISPF**

LOGON SCHEDULER

SPLIT SCREEN

TEXT EDITOR

BACKGROUND/FORGROUND PROCESSING

PDS PARTITIONED DATA SET

TIME SHARING OPTION MULTIPLE CONCURRENT USERS  
 TO ESTABLISH TSO SESSIONS

ISPF INTERACTIVE SYSTEM PRODUCTIVITY FACILITY

VMAN MULTIPLE SESSION MANAGER BETWEEN TEST REGIONS/  
 ADMNCICS (CONTROL-D)

CAN EXECUTE CLISTS AND REXX EXECs

BROWSE/EDIT DATASETS

PDF PROGRAM DEVELOPMENT FACILITY

ISPF DIALOGS/MENUS

SPF (STRUCTURED PROGRAMMING FACILITY) EDIT COMMANDS  
 INSERT/DELETE/DUPLICATE/REARRANGE/LOCATE

MEMBER SELECTION MENU/PF KEYS

SDSF SYSTEM DISPLAY AND SEARCH FACILITY  
 COMMUNICATE AND CONTROL OPERATIONS ON MVS/JES2 SYSTEM

**FILEAID/DATA-XPRT****COMPUWARE**

LOCATE/MANIPULATE/ACCESS DATA IN ENTERPRISE ENVIRONMENT

RDX dataager

CREATE/FORMAT/LOAD/TRANSFER/CONVERT

PRINT RECORDS IN THREE DIFFERENT FORMATS

COBOL OR PL/1 RECORD LAYOUTS AS TEMPLATES

PROCESS FILES IN ALL ACCESS METHODS:

QSAM	QUEUED SEQUENTIAL ACCESS METHOD I/O RECORDS PLACED IN A QUEUE TO DECREASE I/O TIME.
ISAM	INDEXED SEQUENTIAL ACCESS METHOD KEY FIELD IN RECORD AND INDEX
UNSPANNED BDAM	BASIC DIRECT ACCESS METHOD USED FOR PROGRAM STORAGE/QUEUE XFER
VSAM	VIRTUAL STORAGE ACCESS METHOD
KSDS	KEY SEQUENC DATA SET READ SEQUENTIALLY OR BY KEY
ESDS	ENTRY SEQUENCED DATA SET READ IN ENTRY SEQUENCE OR BY RELATIVE BYTE ADDRESS
RRDS	RELATIVE RECORD DATA SET REQD SEQUENTIALLY OR BY KNOWING RECORD NUMBER
BPAM	PDS

REFORMATS RECORDS

ACCUMULATES TOTALS

FUNCTIONS:

COPY  
DROP  
DUMP  
LIST  
PRINT  
TALLY  
UPDATE

PARAMETERS:

ACCUM  
REPLACE  
IF  
MAXOUT  
MOVE  
RBA

BEGIN AT RELATIVE BYTE/BLOCK ADDRESS

OPERATORS: EQ,NE,GT,LT,GE,LE

**4. IMS**

INFORMATION MANAGEMENT SYSTEM/VIRTUAL STORAGE

IQF	INTERACTIVE QUERY FACILITY
MSC	MULTIPLE SYSTEM COUPLING
RSR	REMOTE SITE RECOVERY

RLT RECOVERY LEVEL TRACKING  
 DLT DATABASE LEVEL TRACKING  
 DCE DISTRIBUTED COMPUTING ENVIRONMENT  
 MPR MESSAGE PROCESSING REGION  
 BMP BATCH MESSAGE PROCESSING  
 DBRC DATA BASE RECOVERY CONTROL

SEGEENT UNIT OF ACCESS MADE OF ONE OR MORE FIELDS OF DATA

KEYED LINKAGE POINTERS ESTABLISH SEGMENT RELATIONSHIPS

RACF RESOURCE ACCESS CONTROL FACILITY

FAILURE RECOVERY  
 CHECKPOINTS  
 MESSAGE QUEUES  
 EMERGENCY RESTARTS  
 LOGS  
 AUTOMATIC BACKUPS  
 UTILITIES

PCB MASK  
 DATA BASE NAME 8  
 SEGMENT LEVEL 2  
 STATUS CODE 2  
 PROCESS OPTION 4  
 SEGMENT 8  
 KEY LENGTH 4  
 FEEDBACK AREA VARIABLE

IMS VS APPLICATIONS  
 MPP MESSAGE PROCESSING PROGRAMS  
 MP MESSAGE DRIVEN PROGRAMS  
 BMP BATCH MESSAGE PROGRAMS/JCL

PARENT/ROOT  
 CHILD/DEPENDENT  
 TYPE/OCCURANCE  
 TWIN - MULTIPLE OCCURANCE

SEQUENCE IS ALWAYS TOP TO BOTTOM AND LEFT TO RIGHT

		1	
	2	5	7
3	4	6	

VIRTUAL STORAGE ALLOWS A COMPUTER USER TO ACT AS IF MORE COMPUTER MEMORY IS AVAILABLE THAN THERE ACTUALLY IS BY SWAPPING PROGRAMS AND DATA BETWEEN PERIPHERAL/AUXILIARY MEMORY AND COMPUTER MEMORY

DBMS DATABASE MANAGEMENT SYSTEM

INCREASE DATA INDEPENDENCE/INTEGRITY/SECURITY  
 DECREASE REDUNDACY  
 PROVIDE DC FACILITIES

HIERARCHIAL IMS DL/1 IBM  
 INVERTED TREE STRUCTURE/PARENT AND CHILD WITH AN EMBEDDED LINK (POINTER/PATH ACCESS)  
 NETWORK IDMS, TOTAL, ADABAS  
 RELATIONAL DB2/SQL

DB FACILITIES FOR DEFINING, CREATING, MAINTAINING IMS/VS DATA BASE (DL/1) AND TO PROCESS BATCH

DL/1 INPUT/OUTPUT LANGAUGE WHICH ALLOWS PROGRAMS TO

## TRANSFER DATA:

GET/INSERT/DELETE/REPLACE

DML DATA MANIPULATION LANGUAGE COMMANDS/DL/1  
RETRIEVAL GU GET UNIQUE

GHU HOLD

GHNP NEXT, WITHIN PARENT

ALTER REPL

ADD ISRT

DELETE DLET

## IDMS

RETRIEVAL FIND/GET

ALTER MODIFY

ADD STORE

DELETE DELETE

## SQL

RETRIEVAL SELECT

ALTER UPDATE

ADD INSERT

DELETE DELETE

DC DATA COMMUNICATION PROVIDES TRANSMISSION BETWEEN IMS AND  
REMOTE TERMINALS. SUPPORT TELEPPROCESSING NETWORK

IQF INTERACTIVE QUERY FACILITY TERMINAL ENTERED QUERIES

FP FAST PATH MESSAGE PROCESSING

DBD DATA BASE DEFINITION

## DATA BUILDING BLOCKS/IMS:

FIELD

SEGMENT

PHYSICAL DB

## IDMS:

DATA ITEM

RECORD

SET

## DB2:

ATTRIBUTE/COLUMN

TUPLE/ROW

RELATION/TABLE

## BATCH AND ONLINE

DEVELOP AND RUN DB/CD APPLICATIONS VIA A HIGH-LEVEL APPLICATION  
PROGRAMING INTERFACE, A PROGRAM SCHEDULING FACILITY, AND  
A COMPREHENSIVE SET OF UTILITY PROGRAMSLOGICAL STRUCTURES ARE LINIMITED TO HIERACHIES AND ARE  
LARGELY INDEPENDENT OF UNDERLYING DATA ORGANIZATIONS  
AND STORAGE DEVICES

## RESTART, RECOVERY, AND REORGANIZATION FACILITIES

SEGMENT UNIT OF ACCESS

HIERACHIAL STRUCTURE PARENT/CHILD RELATIONSHIPS

ROOT SEGMENT NO PARENT

ANALOGY  
 PHONE BOOK  
     WHITE PAGES                      PHYSICAL  
     YELLOW                              LOGICAL

HSAM    HIERARCHIAL SEQUENTIAL ACCESS METHOD  
 HIDAM   MOST EFFICIENT FOR EQUAL AMOUNT OF DIRECT/SEQ

          USE TWIN BACKWARD POINTERS ON ROOT SEGEMENTS

HDAM    HIERARCHIAL DIRECT ACCESS METHOD/ESDS VSAM  
 HISAM   HIERACHRIAL INDEXED ...  
 GSAM    GENERALIZED SEQUENTIAL ACCESS METHOD/  
           IMS COMPATIBLILITY WITH VSAM

WHICH ACCESS METHOD/DATA BASE CRITERIA:  
 ROOT ACCESS REQUIREMENTS  
 INSERT/DELETE RECORD REQUIREMENTS  
 DEPENDENT ACCESS REQUIREMENTS  
 FILE GROWTH/SPACE REQUIREMENTS

FAST PATH        dB's USED FOR FAST PROCESSING OF SIMPLE  
                   DATA STRUCTURES/REQUIRES IMS/VS DC

          DEDB                      DATA ENTRY DATA BASE  
           MSDB                      MAIN STORGAE DATA BASE

DATA BASE DEFINITION:

DBD                      DATA BASE DESCRIPTION/CONTROL BLOCK DESCRIBES  
                           PHYSICAL STRUCTURE OF DB  
                           LOGICAL AND PHYSICAL  
                           MAPS TO STORAGE BY ENTERING PARMS TO ESTABLISH RELATIONSHIP

                          DATA BASE NAME/SEGMENT LEVEL/STATUS CODE ETC.

PSB                      PROGRAM SPECIFICATION BLOCK/DEFINES APPLICATION  
                           PROGRAM'S VIEW (S) OF DB

                          DEFINES TO DLI FOR EACH APP THE DB, THE DATA WITH EACH DB, AND  
                           THE OPERATIONS ALLOWED

PCB                      PROGRAM COMMUNICATION BLOCK/  
                           COMMUNICATION BETWEEN IMS AND APPLICATION

PROGRAMING            LOGICAL TERMINAL NAME/MESSAGE OUTPUT  
                           DESCRIPTION, ETC.

ACBGEN APPLICATION CONTROL BLOCK GENERATION

STATUS CODE            STATUS OF I/O OPERATIONS

IMS JCL:

          IMS  
           BMP                      BATCH MESSAGE PROCESSING

CALL FORMAT            CALL 'CBLTDLI' USING  
                           FUNCTION  
                           PCB MASTER DEFININED IN LINKAGE SECTION  
                           I/O AREA/SEGENMENT  
                           SSA    0-15 SEGMENT SEARCH ARGUMENTS

DEBUGGING:            DLI TRACE

I/O:  
           RANDOM  
           SEQUENTIAL



COLLECTION OF TABLES MAS SIZE 2 GB (FILES) CONSISTING OF ROW MAX LENGTH 4005 BYTES(RECORDS) AND COLUMNS MAX IS 255(FIELDS) EACH COLUMN/ROW CAN CONTAIN ONLY ONE DATA VALUE

RUNS AS SUBSYSTEM UNDER MVS

#### DataBase

Base Table  
View  
Domain all possible values for a column  
Primary key  
Foreign Key identifies a column which ids a primary key of a different table  
Reduces reducendency  
Allows extendability

#### Relational operators

Union LIKE  
Intersection  
DifferenceNOT  
Permutation  
Division  
Selection  
Projection  
Join UNLIKE

PARENT TABLE TABLE THAT CONTAINS COLUMN OF UNIQUE VALUES THAT ARE DUPED IN OTHER

#### KEYS

PRIMARY KEY COLUMN OF UNIQUE VALUES  
FOREIGN KEY KEY VALUES FROM PARENT

ACCESSED WITH SEQUENTIAL QUERY LANGUAGE (SQL) STATEMENTS

REPORTS/COUNTS/BREAK POINTS/GROUPING

DB2 DATA BASE  
EXEC SQL  
BINDS/DCLGENS (UTILITY TO GENERATE STATEMENTS)

SQL SEQUENTIAL QUERY LANGUAGE (SEE-QWELL)  
STATIC/DYNAMIC SQL  
PROVIDES DATA DEFINITION/MANIPULATION/AUTHORIZATION FACILITIES  
SQL STATEMENTS  
CLOSE  
COMMIT  
CREATE  
DELETE  
DECLARE  
DROP  
EXECUTE  
EXPLAIN see what options for execution DB2 has chosen  
FETCH  
INCLUDE  
INSERT  
LOCK TABLE  
OPEN

SET-LEVEL DATA MANIPULATION STATEMENTS  
TYPICALLY OPERATE ON A TABLE RATHER THAN A

ROW/EMBEDDED SQL STATEMENTS (FETCH) CAN  
PROCESS ONE ROW AT A TIME

DATA DEFINITION:

DECLARE  
    DECLARE DATA DIVISION DESCRIPTIONS  
    BY USING SQL DECLARE TABLE  
DCLGEN

EMBEDDED SQL            CURSOR  
    USED TO PROCESS SELECTED DATA ONE ROW AT A TIME

COMMUNICATION AREA    SQLCA  
    IN WORKING -STORAGE SECTION  
    EXEC SQL  
        INCLUDE SQLCA  
    END-EXEC.  
    ALLOWS COMMUNICATION BETWEEN  
    DB2 AND APPLICATION PROGRAMS  
    RETURN CODES    0            SUCCESSFUL  
                                  POSITIVE        EXCEPTION  
                                  NEGATIVE        ERROR CONDITION

EXCEPTIONAL CONDITIONS:

WHENEVER  
SQLCODE

SQL SELECT OPTIONS:

JOIN  
    SELECT DATA FROM TWO TABLES based on value of one or more  
    common tables  
    NAME BOTH TABLES IN THE FROM CLAUSE/SPECIFY  
    THE TWO COLUMNS (ONE FROM EACH TABLE) IN THE  
    WHERE CLAUSE

SELECT ca-t1 cb-t1  
    FROM T1 T2  
    WHEN CCT1 = CCT2

MERGE

USE TWO OR MORE SELECT STATEMENTS WITH UNION  
KEYWORD

UNION                    DROP DUPES  
UNION ALL                KEEP DUPES

SELECT \*                DON'T USE/ALL COLUMNS  
    COLUMNS-NAME        SOME COLUMN  
    WHERE                 ROWS  
    ORDER BY             IN ORDER  
    DISTINCT             NON-DUPES  
    BETWEEN/AND/OR        CONDITIONAL

DEBUGGING:

DATA MANIPULATION OPERATIONS:

SELECT  
    RETRIEVE DATA FROM TABLES  
    ROWS                 WHERE  
    ROWS IN ORDER    ORDER BY  
    NO DUPLICATE ROWS   DISTINCT  
    CONDITIONS         BETWEEN/IN/LIKE  
    RETRIEVE INTO PGM    DECLARE CURSOR FOR  
                          SELECT

UPDATE  
DELETE

INSERT  
 CREATE                   DEFINES TABLE/VIEW  
 ALTER                    CHANGE A DESCRIPTION

REVOKE/GRANT   DB PRIVILEGE

DB2PM   DB2 PERFORMANCE MONITOR

DBEDIT   utility for quick construction of DB2 tables

UNDER CICS ONLINE APPLICATIONS MAY READ DB2 DATA USING SQL

UNDER IMS USE DL/1 AND VSAM FILES

DXT       DATA EXTRACT

DB2 INTERACTIVE

DB2I                DB2 INTERACTIVES TO DEVELOP AND RUN APPLICATION  
 PROGRAMS/RUNS UNDER TSO/ISPF

                  PRECOMPIATION PROCESS PRODUCES A DATA BASE  
                   REQUEST MODULE (DBRM) FOR BIND

                  BIND ESTABLISHES A RELATIONSHIP BETWEEN PROGRAM AND  
 DB2 DATA/VALIDATES SQL STATEMENTS/CHECKS  
 AUTHORIZATION/BUILD AND STORES AN APPLICATION  
 PLAN TO ALLOW DB2 TO ACCESS DB2 DATA WHEN  
 PROGRAM EXECUTES

                  SPUFI                SQL PROCESSOR USING FILE INPUT  
                   RUN UNDER ISPF ALLOWING USER  
                   TO ENTER SQL COMMANDS FROM  
                   TERMINAL

                  DCLGEN PL/1 OR COBOL COPYBOOKS  
 BIND  
 RUN

DB2 OBJECTS

DATA BASE  
 TABLE OR INDEX SPACE  
 STORAGE GROUP  
 TABLE  
 INDEX  
 VIEW                PORTION OF ONE OR MORE TABLES/CREATE VIEW

QMF                QUERY MANAGEMENT FACILITY

INTERACTIVE PRODUCT UNDER TSO TO CREATE REPORTS AND CHARTS

EXECUTE A SQL QUERY

END USERS ENTER QUERIES IN SQL OR IN  
 QUERY BY EXAMPLE

SQL COMMUNICATION ARE A FEEDBACK BETWEEN  
 APPLICATIONS PROGRAM AND DB2

SQL QUERY

QBE QUERY

FORMS/CHARTS/UPDATING DATA

COMMANDS  
 DISPLAY  
 LIST  
 EDIT  
 GRANT

## EXPORT

WORKS WITH INTERACTIVE CHART UTILITY TO CREATE BAR, PIE, LINE CHARTS

DON'T USE CONTESIOAN PRODUCTS/SELECT \*/JOINS ON NON-INDEXED COLUMNS

DXT DATA EXTRACT/UTILITY TO EXTRACT DATA FROM IMS  
DATA BASES/VSAM/FOR USE BY DB2 LOAD UTILITIES

LOAD DB2 TABLE VIA DB2 LOAD UTILITY/SQL INSERT STATEMENT

DB2 CATALOG COLLECTION OF TABLES CONTAINING INFO ABOUT DB2 OBJECTCS

SQL STRUCTURED QUERY LANGUAGE  
DDL DATA DEFINITION LANGUAGE DEFINE OBJECTS  
CREATE  
ALTER  
DROP  
DML DATA MANIPULATION LANGUAGE ACCESS AND UPDATE DATA IN TABLES  
SET-LEVEL LANGUAGE MEANS THAT WHEN A REQUEST IS MADE A SET OF ROWS  
ACTED ON THAT REQUEST (VS. PROCEDURAL)  
RETRIEVAL  
SELECT CIOLUMNS FROM TABLES WHERE CONDITIONS  
INSERT/DELETE/UPDATE  
GROUP BY/ODER BY  
HAVING  
BUILT IN FUCNTIONS  
MIN MAX SUM COUNT AVG  
OTHER OPERATORS  
UNION  
DISTINCT  
BETWEEN  
DCL DATA CONTROL LANGAUGE TO ASSIGN AND REVOKE AUTH PRIVILEGES

APLICATION PROGRAMS CONTAIN EMBEDDED SQL (STATIC SQL) APPLICATION-EMBEDDED  
SQL

BIND THE DB2 PROCESS THAT  
TAKES INPUT FROM DB2  
PRECOMPILE AND  
CONVERTS IT TO AN APP  
PLAN

PLAN STRUCTURE USED BY DB2  
THAT CONTROLS ACCESS  
TO DATA AND SPECIES  
THE PATH TO BE  
FOLLOWED

DISTINCT FREQUENCY COUNT

1. DCLGEN
2. PRECOMPILE CREATES DATA BASE REQUEST MODULE DBRM
3. BIND CREATES APPLICATION PLAN
4. COMPILE AND LINK
5. EXECUTION

DEADLOCK UNRESOLVED CONTENSION

HOST VARIABLE A VARIABLE IN A HOST PROGRAM REFERRREDNCED BY EMBEDDED SQL

SYNONYM AN ALTERNATIVE NAME FOR A DB2 TABLE OR VIEW

DB2 CATALOG

SYSIBM.SYSCOLAUTH	COLUMN UPDATE PRIVILEGES
SYSDBRM	ONE ROW FOR EACH DBRM
SYSVIEWDEP	VIEW DEPENDENCY INFORMATION

DYNAMIC SQL    SUCH AS DB2I OR QMF COMPILED AT EXECUTION TIME

SPUFI    SQL PROCESSING USING FILE INPUT            SQL STATEMENTS EXECUTED DIRECTLY

PRECOMPILE/COMPILE/BIND/EXECUTE

DBRM    DATA BASE REQUEST MODULE

JOINING TABLES

INNER JOIN	IF NO MATCH IS FOUND THE ROW IS ELIMINATED
OUTER JOIN	ALL DATA INCLUDING THOSE WITHOUT MATCHES/UNION

EX:

```

Select A.CO
      A.ACCT
      B.NAME
      B.ACCT
FROM TABLE A, TABLE B
WHERE A.ACCT = INPUT
AND A.ACCT = B.ACCT

```

COBOL

```

SQLCA 01 IN DATA DIVISION
EXEC SQL IN PROCEDURE DIVISION

```

DECLARE TABLES

FETCH CURSOR    MULTIPLE ROWS RETURNED

SINGLE ROW SELECT/WHERE

BASE TABLE    THE PHYSICALLY STORED DATA IN A RELATIONAL SYSTEM

BIND            OUTPUT FROM THE DB2 PRECOMPILER IS CONVERTED TO A USUABLE CONTROL  
STRUCTURE CALLED AN APPLICATION PLAN

BUNDLED PRIVILEGES    SET OF PRIVILEGES ASSIGNED TO A SYSTEM NAME

COLUMN        FIELD

CURSOR        CONTROL STRUCTURE THAT POINTS TO A ROW

DATA VALUE    INTERSECTION OF A ROW AND COLUMN

EMBEDDED SQL    SQL STATEMENTS THAT ARE INCLUDED WITH AN APPLICATION  
PROGRAM

JOIN            RETRIEVAL OF DATA FROM TWO OR MORE TABLES

OBJECT        ANYTHING CREATED OR MANIPULATED BY SQL

QBE            QUERY BY EXAMPLE--            INTERFACING PRODUCT

REFERENTIAL INTEGRITY    MAINTAIN SAME FIELD IN MULTIPLE TABLES

SET-LEVEL LANGUAGE    PROCESSING LANGUAGE THAT ACTS ON SETS OF ROWS RATHER THAN  
A SINGLE RECORD

SQL

DDL	DATA DEFINITION LANGUAGE USED TO DEFINE OBJECTS		
	CREATE		
	STOGROUP	A	D
	DATABASE		D

	TABLESPACE	A	D
	TABLE	A	D
	SYNONYM		D
	INDEX	A	D
	VIEW		D
	ALTER		
	DROP		
DML	DATA MANIPULATION LANGUAGE USED TO ACCESS AND UPDATE DATA IN TABLES		
	SELECT		
	INSERT/DELETE/UPDATE		
	GROUP BY/HAVING		
	ORDER BY		
	MIN/MAX/SUM/AVG/COUNT		
DCL	DATA CONTROL LANGUAGE USED TO ASSIGN AND REVOKE PRIVILEGES TO AUTH ID		
	GRANT/REVOKE		
SPUFI	SQL PROCESSING USING FILE INPUT		
DB2 UTILITIES			
	LOAD		
	COPY		
	MERGE COPY		
	RECOVER		
	REORG		
	RUNSTATS		
	STOSPACE		
SQL			
	DDL	DATA DEFINITION LANGUAGE	
		CREATE	
		ALTER	
		DROP	
	DML	DATA MANIPULATE LANGUAGE	
		SELECT	
		INSERT/DELETE/UPDATE	
		GROUP/ORDER BY	
	DCL	DATA CONTROL LANGUAGE	
		GRANT	
		REVOKE	
	EMBEDDED/STATIC SQL IN APPLICATION PROGRAMS		
	DYNAMIC SQL VIA DB2I OR QMF		
	BETWEEN		
	NOT IN/IN		
	LIKE	PARTIAL CONDITION	
	NULL		
	HAVING		
	INSERT		
	UPDATE		
CORRELATION VARIABLES	ALIAS NAMES ASSIGNED TO TABLES		
CORRELATED SUBQUERIES	SUBQUERY EXECUTED IN CORRELATION		
WITH	THE MAIN QUERY		
SUBQUERY	SELECT/FROM/WHERE EXPRESSION NESTED		
	WITHIN ANOTHER SELECT		
HOST VARIABLES	QUALIFIED WITH NAME OF THE STRUCTURE		
NULL INDICATOR VARIABLE ½ WORD	INTERGER VARIABLE TO INDICATE		
	IF COLUMN CONTAINS DATA OR IS NULL		
COMMIT PROCESSING	CURSORS ARE CLOSED AND DATABASE CHARGES		
	ARE MADE PERMANENT		
BIND	PROCESS BY WHICH THE OUTPUT FROM THE DB2 PRECOMPILER IS CONVERTED		
	TO A USABLE CONTROL STRUCTURE CALLED AN APPLICATION PLAN		
CURSOR	CONTROL STRUCTURED USED BY APP TO POINT TO A ROW		
	IF MULIPLE ROWS ARE RETURNED THE APPLICATION MUST		
	PROCESS THE RESULT WITH FETCH CURSOR		

DCLGEN            COMPONENT OF DB2 THAT GENERATE SQL TABLE DECLARATIONS AND  
COBOL DATA STRUCTURES THAT CONFORM TO THE TABLE

JOIN             RELATIONAL OPERATION THAT ALLOWS RETRIEVAL OF DATA FROM TWO  
OR MORE TABLES BASED ON MATCHING CONTROL VALUES

UNION            SQL OPERATION THAT COMBINES THE RESULT OF TWO SELECT  
STATEMENTS

CASCADE         DELETION OF A PARENT ROW WILL CAUSE ALL DEPENDENT TABLE ROWS  
WITH A MATCHING FOREIGN KEY TO BE DELETED

                  WHEN A PRIMARY KEY IS UPDATED, THE FOREIGN KEYS MAY ALL BE  
CHANGED AT THE SAME TIME.

RESTRICT        DELETION OF A PARENT ROW WILL NOT OCCUR IF ANY DEPENDENT  
TABLE ROW WITH A MATCHING FOREIGN KEY IS PRESENT

                  WHEN PRIMARY KEY IS UPDATED, FOREIGN KEYS ARE NOT UPDATED

CALL ATTACHMENT FACILITY        BATCH/JCL ACCESS TO DB2

SOURCE CODE WITH EMBEDDED SQL  
PRECOMPILE

SOURCE CODE                      DBRM  
COMPILE                            BIND/DB2 CATALOG  
LINKEDIT                          APPLICATION PLAN  
LOAD MODULE WITH DB2 INTERFACE

DISTINCT         ELIMINATION OF DUPLICATE ROWS

ORDER BY ASC/DESC

INNER JOIN       compares the contents of the join field in each row to the contents of the join field in  
                  the second table

OUTER JOIN       JOIN NEEDED FOR ALL DATA INCLUDING THOSE WITHOUT MATCHES  
                  There is no sql statmenty that directly produ ces an outer join. Instead the functions  
                  of union and subqueries muts be used

Subquery         select from where expression nested within another select

APPLICATION STRUCTURE  
IDENTIFICATION DIVISION  
ENVIRONMENT DIVISON  
DATA DIVISION  
                  01 SQLCA  
                  01 TABLE DECLARATIONS  
                  01 HOST VARIABLES  
PROCEDURE DIVISON  
                  EXEC SQL

RECOVERY UTILITIES  
IMAGE COPY UTILITIES  
                  FULL/INCREMENTAL/MERGE  
APPLYING DATA BASE CHANGES  
                  ICREST  
                  ICMERG  
                  LOGAPPLY  
INDEX MAINTENANCE  
                  CHECK INDEX  
                  RECOVER INDEX