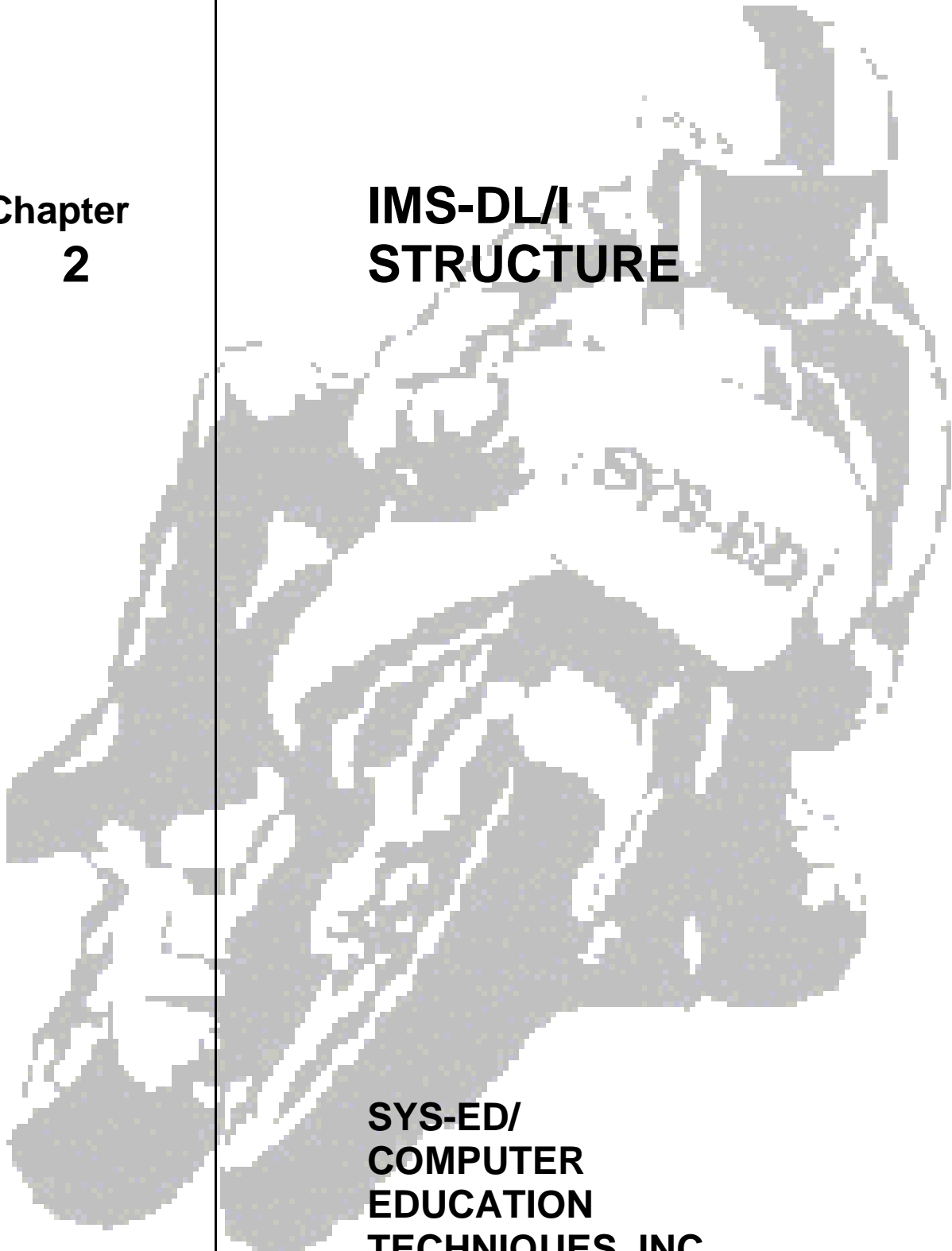


**Chapter  
2**

**IMS-DL/  
STRUCTURE**

**SYS-ED/  
COMPUTER  
EDUCATION  
TECHNIQUES, INC.**



**Objectives**

You will learn:

- C Database hierarchy.
- C DL/I database segments.
- C DL/I database records.
- C Parent and child segments.
- C Twins and siblings.
- C Segment limits and sensitivity.
- C Processing options.
- C Key (sequence) field.
- C Search field.
- C DBDGEN process.
- C PSBGEN process.

## 1 Database Hierarchy

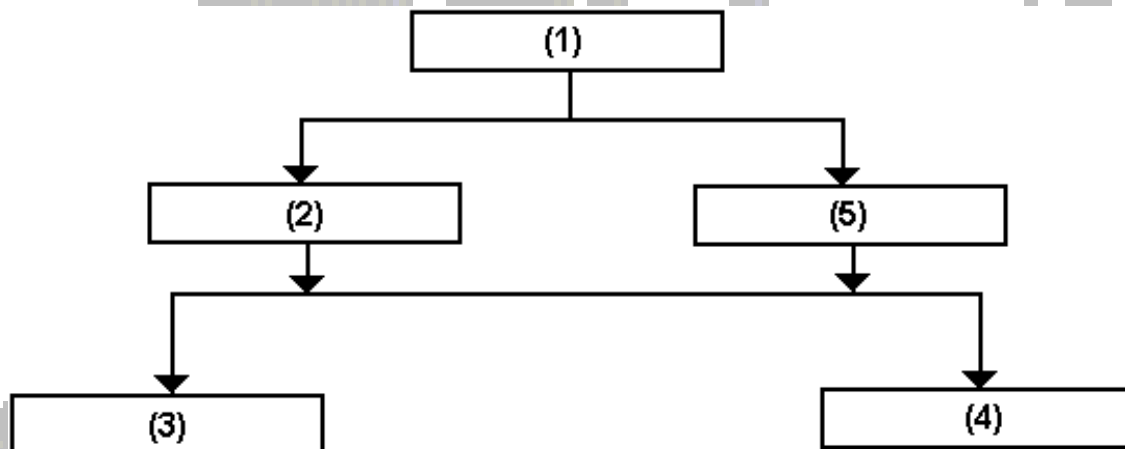
An IMS database record is broken down into segment types. The segment types are organized in a hierarchy.

A hierarchical structure describes relationships between segment types.

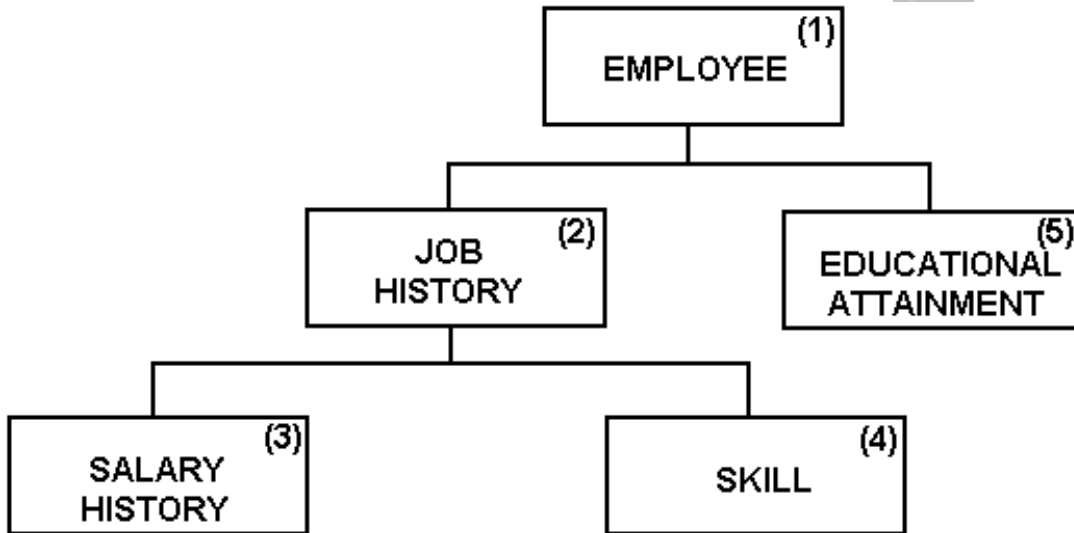
Within IMS, the following rules apply:

- C The highest segment in the hierarchy is the root segment. There is only one root segment type.
- C There is only one path to any given segment.
- C The segment types are numbered from top to bottom, then left to right.

The database hierarchy chart is a graphic representation of a hierarchical structure.



1.1 Database Hierarchy Chart



---

## 2 DL/I Database Segments

A segment is the smallest unit of data which can be retrieved from a data base.

- C The first segment is the root segment.
- C A segment consists of a control information and one or more data fields.

A segment type identifies a particular segment such as a job history segment, salary history segment, etc.

- C There may be a maximum of 255 segment types.

A segment occurrence is a specific instance of a segment type.

- C For example, an employee may have several occurrences of the salary history segment, one segment for each salary level.

SEGMENTS

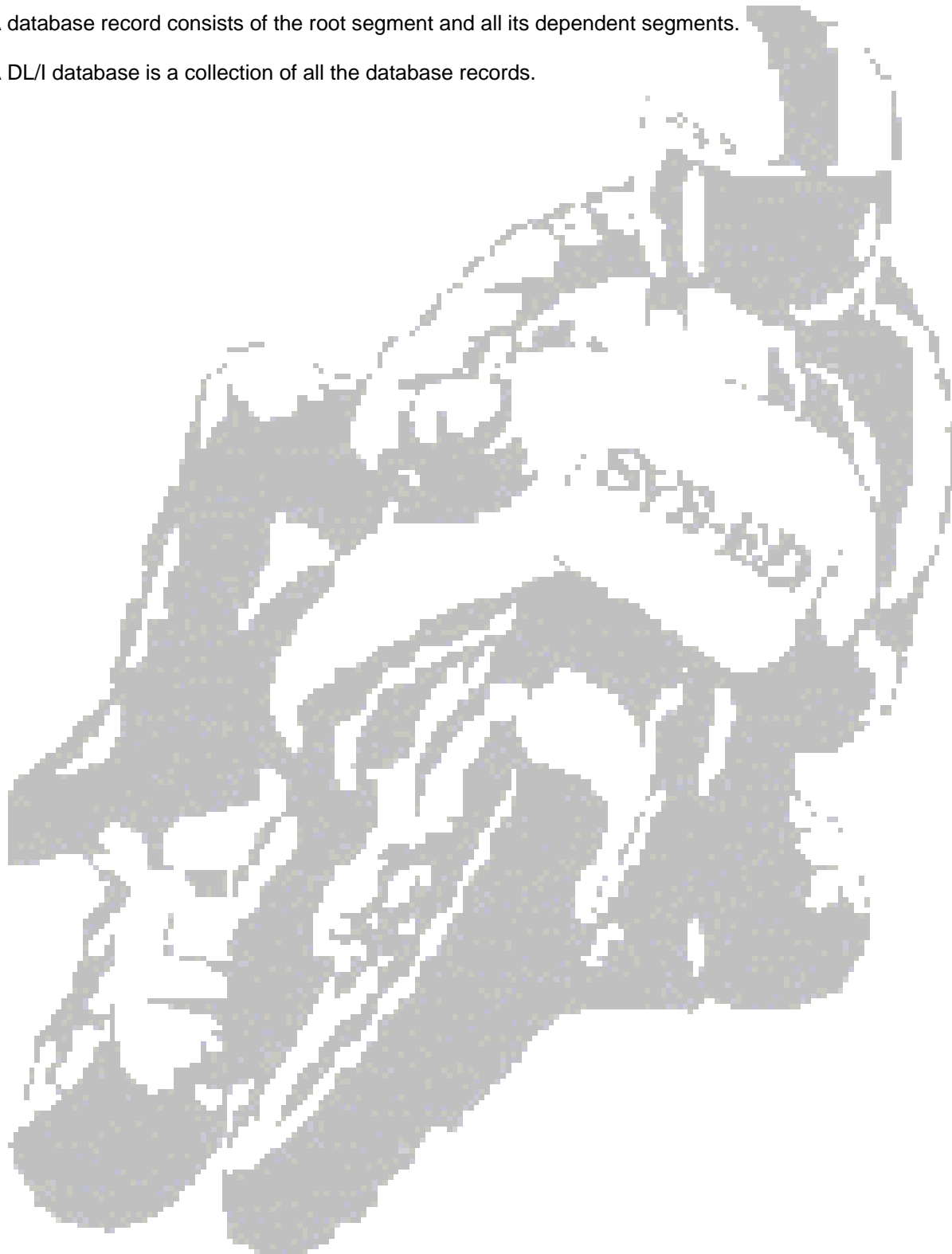
		*	
S	D	P A P	SEGMENT DATA
E	E	O D O	
Q	L	S D I	
	E	S R N	
I	T	I E T	
D	E	B S E	
		L S R	
		E S	

---

### **3 DL/I Database Records**

A database record consists of the root segment and all its dependent segments.

A DL/I database is a collection of all the database records.



---

## **4 Parent and Child Segments**

A child is a dependent segment of some higher level segment.

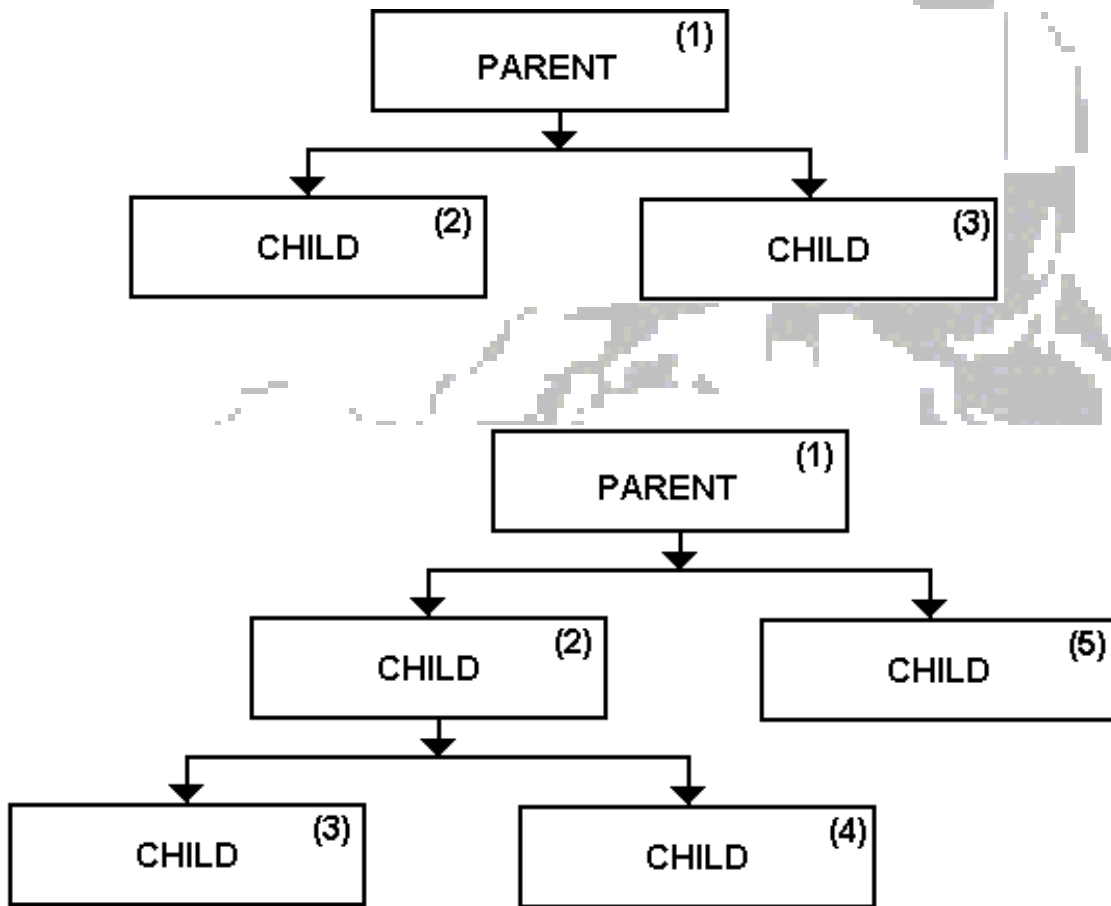
A child's higher level segment is called a parent.

A parent may have many children, but a child may have only one physical parent.





PARENT/CHILD SEGMENT



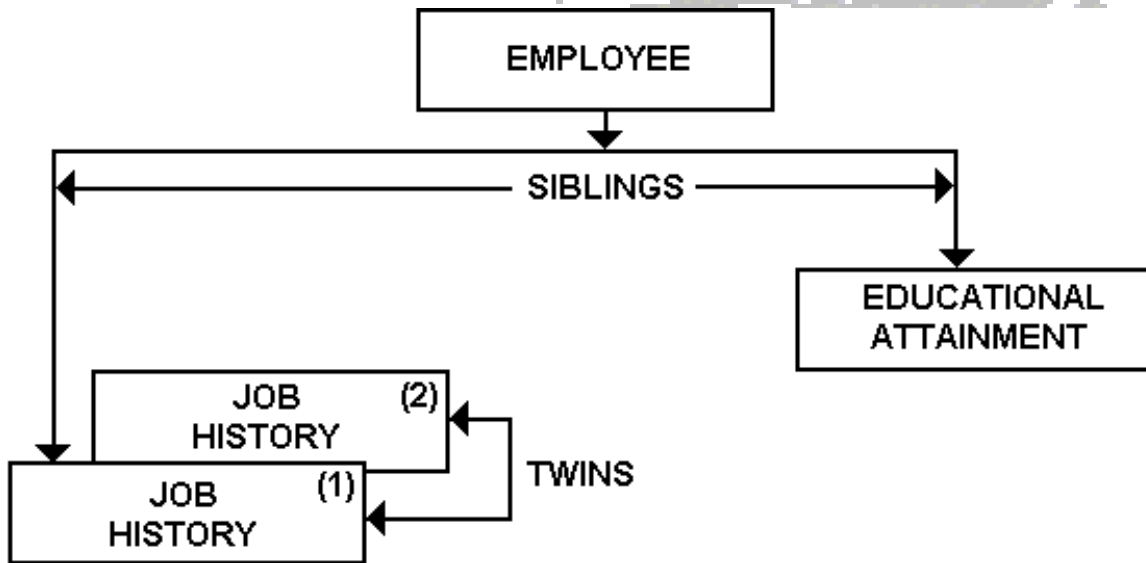
**5 Twins and Siblings**

Twin segments are all occurrences of a particular segment type under a single parent.

C For example, all occurrences of the job history segment type are twins.

Sibling segments are occurrences of different segment types under a single parent.

C For example, segment types of job history and educational attainment are siblings.



---

## 6 Segment Limits and Sensitivity

- C Maximum of 255 segment types.
- C Maximum of 15 segment levels.
- C No limit on segment occurrences (except disk capacity).
- C If VSAM is the operating system access method, then the segments may be of variable length.
- C The PCB specifies which segments a program may access (ie. which segments are sensitive).
- C Root segments are always sensitive.

---

## 7 Processing Options

These are actions that a program may perform on segments.

A security mechanism specifies how data can be processed.

The most frequently used options are:

G	Read.
I	Insert.
R	Replace. The G option is implied.
D	Delete. The G option is implied.
A	All options allowed.
K	Access key only.

Additional options for on-line programs:

E:	Exclusive use of hierarchy or segment.
GO:	Current segment may be used by other programs and may also access segments currently being updated by those programs.

---

## 8 Key (Sequence) Field

A key field is used by DL/I to keep segments in ascending sequence.

- C The employee number in the employee segment type is an example of such a field.
- C Key fields are not required in DL/I.
- C Up to one key field per segment may be specified for a primary index; secondary indexes may reference multiple fields.
- C If a key field is defined as unique, DL/I will not allow twins to have the same key value.
- C A key field may be sensitive (ie. accessible) even if its data portion is not.

---

## **9 Search Field**

A search field is used by DL/I to locate particular segments.

For example, account balance could be defined as a search field and then for determining which accounts have an outstanding balance of at least \$10,000.

There is maximum of 255 search fields and key fields per segment.

A search field is not a key field.

---

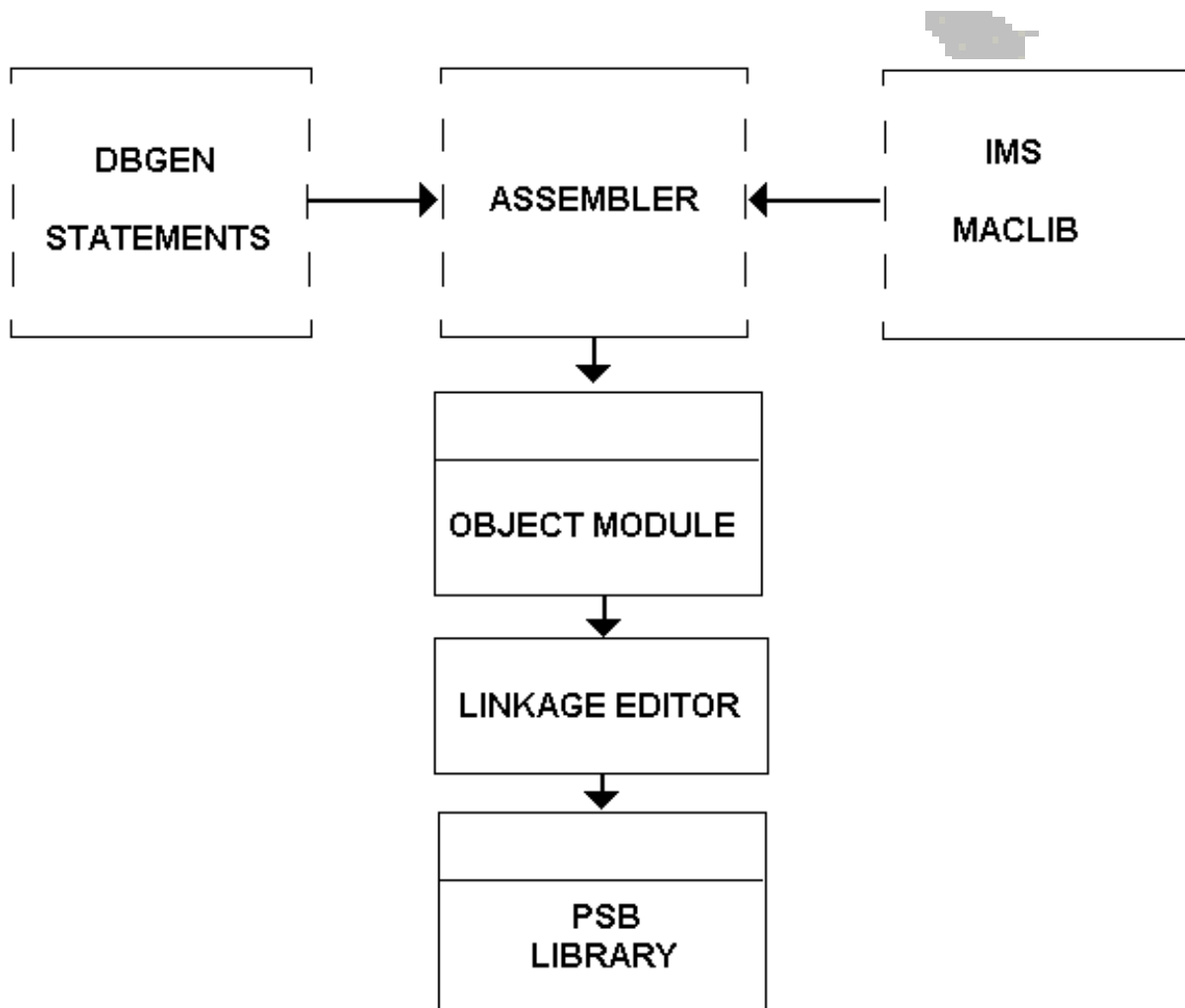
## **10 DBDGEN Process**

The DBDGEN is a set of Assembler statements that generates the DBD.

The DBDGEN is:

- C Coded by the database administrator.
- C Assembled and linked to the IMS-DL/I load library.
- C Used by all programs accessing the database.

DBGEN PROCESS





---

### **10.1 DBDGEN Example**

The DBDGEN gives physical information about a database such as:

- C Database name
- C Access method used
- C Segment name and length
- C Hierarchical sequences
- C Key and search fields

```
DBD NAME=CUSTAR,ACCESS=HISAM
DATASET DD1=PRIME,OVFLW=OVERFLOW,DEVICE=3330
SEGM  NAME=CUSTINFO,PARENT=0,BYTES=60
FIELD NAME=(CUSTNO,SEQ,U),BYTES=5,START=1,TYPE=C
FIELD NAME=CUSTNAME,BYTES=20,START=6,TYPE=C
FIELD NAME=CUSTADDR,BYTES=15,START=26,TYPE=C
FIELD NAME=CITYST,BYTES=15,START=41,TYPE=C
FIELD NAME=ZIP,BYTES=5,START=56,TYPE=C
SEGM  NAME=SHIPADDR,PARENT=CUSTINFO,BYTES=40
FIELD NAME=(SHPCSTN,SEQ,U),BYTES=5,START=1,TYPE=C
FIELD NAME=SHPCZIP,BYTES=5,START=36,TYPE=C
SEGM  NAME=BILLING,PARENT=CUSTINFO,BYTES=22
FIELD NAME=BILLINV,BYTES=8,START=7,TYPE=C
SEGM  NAME=CASH,PARENT=CUSTINFO,BYTES=22
FIELD NAME=CASHINV,BYTES=8,START=7,TYPE=C
DBDGEN
FINISH
END
```

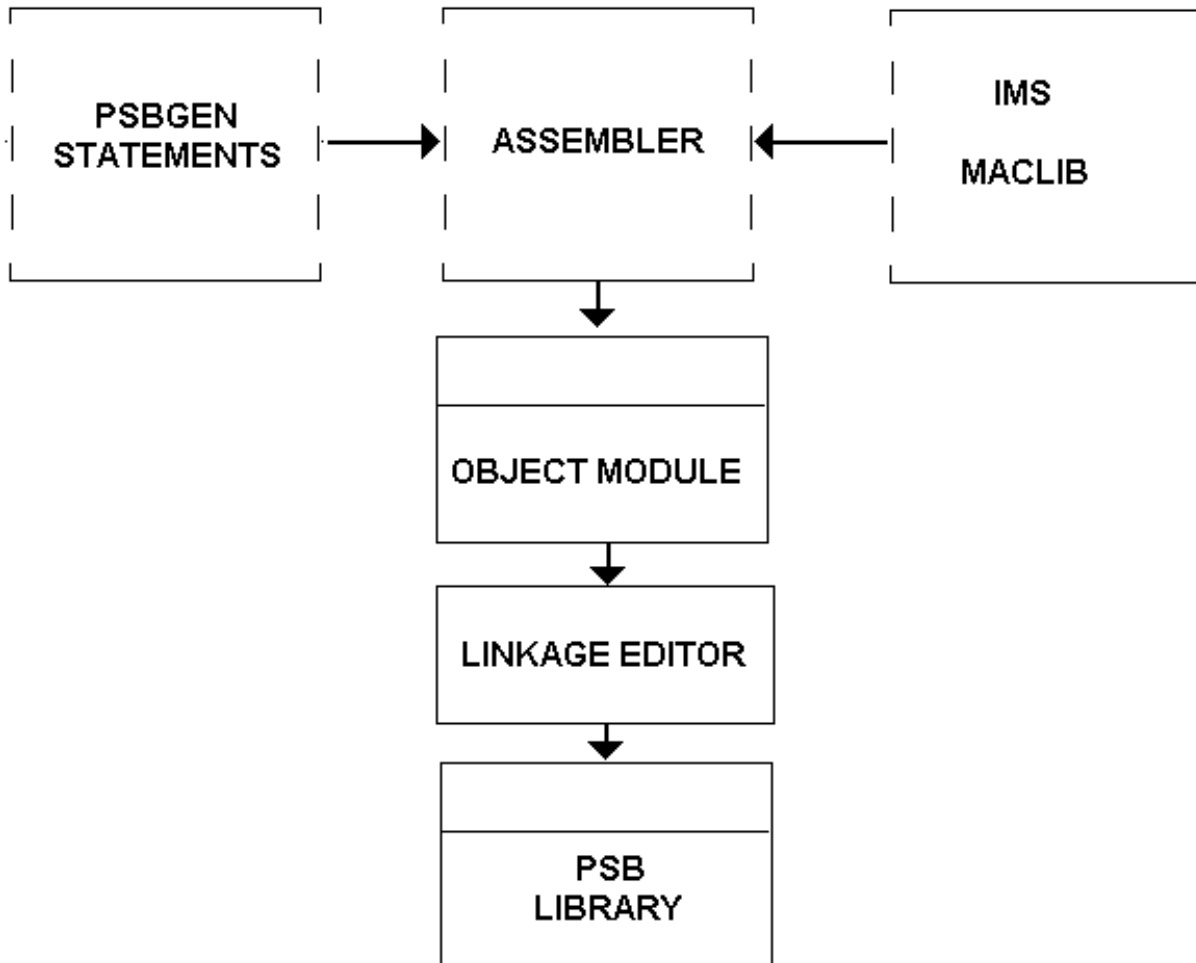
---

## 11 PSBGEN Process

The PSBGEN is a set of assembler statements which generate one or more PCBs.

The PSBGEN (Program Specification Block Generation) is similar to the DBDGEN.

- C The PSBGEN is coded, assembled, and linked in the same way as the DBDGEN.
- C The set of PCBs within the PSB is an application structure.



---

## 11.1 PCB Examples

PSB - 1:

```
PCB          TYPE=DB,DBDNAME=CUSTAR,PROCOPT=A,KEYLEN=10
SENSEG      NAME=CUSTINFO,PARENT=0
SENSEG      NAME=SHIPADDR,PARENT=CUSTINFO
PSBGEN      LANG=COBOL,PSBNAME=ARPGM01
END
```

PSB - 2:

```
PCB          TYPE=DB,DBDNAME=CUSTAR,PROCOPT=A,KEYLEN=5
SENSEG      NAME=CUSTINFO,PARENT=0
SENSEG      NAME=BILLING,PARENT=CUSTINFO
SENSEG      NAME=CASH,PARENT=CUSTINFO
PSBGEN      LANG=COBOL,PSBNAME=ARPGM02
END
```