

#### Universidad Autónoma de Baja California Sur Departamento Académico de Sistemas Computacionales Ingeniería en Desarrollo de Software

# Programación de Sistemas Análisis Semántico Parte 2

MSC. Martin Agúndez Amador

Para el traductor del lenguaje SQL estas tablas contienen toda la información sobre la estructura de la(s) base(s) de datos a utilizar.

Se tienen 3 tablas de símbolos semánticas: Tablas, Atributos y Restricciones.

El llenado se realiza durante la fase del análisis léxico (escáner).

	TABLAS								
No.	Nombre	No.	No.						
Tabla		Atributos	Restricciones						
1	DEPARTAMENTOS	2	1						
2	CARRERAS	5	2						

	ATRIBUTOS									
No. Tabla	No. Nombre Atributo		Tipo	Longitud	No Nulo					
1	1	D#	CHAR	2	1					
1	2	DNOMBRE	CHAR	6	1					
2	3	C#	CHAR	2	1					
2	4	CNOMBRE	CHAR	3	1					
2	5	VIGENCIA	CHAR	4	1					
2	6	SEMESTRES	NUMERIC	2	1					
2	7	D#	CHAR	2	1					

RESTRICCIONES										
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo				
Tabla	Restricción			asociado						
1	1	1	PK_DEPARTAMENTOS	1	-					
2	2	1	PK_CARRERAS	3	-					
2	3	2	FK CARRERAS	7	1	1				

	TABLAS								
Γ	No. Nombre		No.	No.					
L	Tabla		Atributos	Restricciones					
	1	DEPARTAMENTOS	2	1					
	2	CARRERAS	5	2					

CREATE TABLE DEPARTAMENTOS (
D# CHAR (2) NOT NULL,
DNOMBRE CHAR (6) NOT NULL,

CONSTRAINT PK\_DEPARTAMENTOS PRIMARY KEY (D#));

CREATE TABLE CARRERAS (

C# CHAR(2) NOT NULL,
CNOMBRE CHAR(3) NOT NULL,
VIGENCIA CHAR(4) NOT NULL,
SEMESTRES NUMERIC(2) NOT NULL,
D# CHAR(2) NOT NULL,

CONSTRAINT PK\_CARRERAS PRIMARY KEY (C#),
CONSTRAINT FK CARRERAS FOREIGN KEY (D#) REFERENCES DEPARTAMENTOS(D#));

	ATRIBUTOS									
No. Tabla	No. Atributo	Nombre	Tipo	Longitud	No Nulo					
1	1	D#	CHAR	2	1					
1	2	DNOMBRE	CHAR	6	1					
2	3	C#	CHAR	2	1					
2	4	CNOMBRE	CHAR	3	1					
2	5	VIGENCIA	CHAR	4	1					
2	6	SEMESTRES	NUMERIC	2	1					
2	7	D#	CHAR	2	1					
		•••								

RESTRICCIONES Tipo No. No. Nombre Atributo Tabla Atributo Tabla Restricción asociado PK DEPARTAMENTOS 1 1 1 1 PK CARRERAS 3 2 1 2 2 FK CARRERAS 7 3 1 1

 TABLAS

 No.
 Nombre
 No.
 No.
 No.

 Tabla
 Atributos
 Restricciones

 1
 DEPARTAMENTOS
 2
 1

 2
 CARRERAS
 5
 2

 ...
 ...
 ...
 ...

CREATE TABLE DEPARTAMENTOS (
D# CHAR (2) NOT NOLL,
DNOMBRE CHAR (6) NOT NULL,

CONSTRAINT PK\_DEPARTAMENTOS PRIMARY KEY (D#));

CREATE TABLE CARRERAS (

C# CHAR(2) NOT NULL,
CNOMBRE CHAR(3) NOT NULL,
VIGENCIA CHAR(4) NOT NULL,
SEMESTRES NUMERIC(2) NOT NULL,
D# CHAR(2) NOT NULL,

CONSTRAINT PK CARRERAS PRIMARY KEY (C#),

ATRIBUTOS No. No. Nombre Tipo Longitud No Tabla Atributo Nulo 1 1  $D^{\#}$ CHAR 1 1 DNOMBRE CHAR 6 1 2 3 C# CHAR 2 4 CNOMBRE CHAR CHAR 2 5 VIGENCIA 1 NUMERIC 6 SEMESTRES 1 7 D# CHAR 1

 $\hbox{\tt CONSTRAINT FK\_CARRERAS FOREIGN KEY (D\#) REFERENCES DEPARTAMENTOS (D\#)); } \\$ 

	RESTRICCIONES										
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo					
Tabla	Restricción			asociado							
1	1	1	PK_DEPARTAMENTOS	1	-	-					
2	2	1	PK_CARRERAS	3	-	-					
2	3	2	FK CARRERAS	7	1	1					

	TABLAS									
No. Nombre		No.	No.							
Tabla		Atributos	Restricciones							
, 1	DEPARTAMENTOS	2	1							
2	CARRERAS	5	2							
.\.										

#### CREATE TABLE DEPARTAMENTOS (

D# CHAR(2) NOT NULL, DNOMBRE CHAR(6) NOT NULL,

CONSTRAINT PK\_DEPARTAMENTOS PRIMARY KEY (D#));

#### CREATE TABLE CARRERAS (

C# CHAR(2) NOT NULL,
CNOMBRE CHAR(3) NOT NULL,
VIGENCIA CHAR(4) NOT NULL,
SEMESTRES NUMERIC(2) NOT NULL,
D# CHAR(2) NOT NULL,

		ATRIBUTOS								
	No. Tabla	No. Atributo	Nombre	Tipo	Longitud	No Nulo				
A	1	1	D#	CHAR	2	1				
	1	2	DNOMBRE	CHAR	6	1				
	2	3	C#	CHAR	2	1				
	2	4	CNOMBRE	CHAR	3	1				
	2	5	VIGENCIA	CHAR	4	1				
	2	6	SEMESTRES	NUMERIC	2	1				
	2	7	D#	CHAR	2	1				
[										

CONSTRAINT PK\_CARRERAS PRIMARY KEY (C#),
CONSTRAINT FK\_CARRERAS FOREIGN KEY (D#) REFERENCES DEPARTAMENTOS(D#));

	RESTRICCIONES										
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo					
Tabla	Restricción			asociado							
1	1	1	PK_DEPARTAMENTOS	1	-	-					
2	2	1	PK_CARRERAS	3	-	-					
2	3	2	FK CARRERAS	7	1	1					

	TABLAS									
	No. Nombre		No.	No.						
L	Tabla		Atributos	Restricciones						
	1	DEPARTAMENTOS	<b>→</b> 2	1						
	2	CARRERAS	5	2						
		.:								

CREATE TABLE DEPARTAMENTOS (

CHAR(2) NOT NULL, DNOMBRE CHAR (6) NOT NULL

CONSTRAINT PK\_DEPARTAMENTOS PRIMARY KEY (D#));

CREATE TABLE CARRERAS (

C# CHAR(2) NOT NULL, CNOMBRE CHAR(3) NOT NULL, VIGENCIA CHAR (4) NOT NULL, NUMERIC (2) NOT NULL, SEMESTRES

CHAR(2) NOT NULL,

		ATRIBUT	ros		
No.	No.	Nombre	Tipo	Longitud	No
Tabla	Atributo				Nulo
1	1	D#	CHAR	2	1
1	2	DNOMBRE	CHAR	6	1
2	3	C#	CHAR	2	1
2	4	CNOMBRE	CHAR	3	1
2	5	VIGENCIA	CHAR	4	1
2	6	SEMESTRES	NUMERIC	2	1
2	7	D#	CHAR	2	1

CONSTRAINT PK CARRERAS PRIMARY KEY (C#), CONSTRAINT FK CARRERAS FOREIGN KEY (D#) REFERENCES DEPARTAMENTOS(D#));

	RESTRICCIONES									
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo				
Tabla	Restricción			asociado						
1	1	1	PK_DEPARTAMENTOS	1	-	-				
2	2	1	PK_CARRERAS	3	-	-				
2	3	2	FK CARRERAS	7	1	1				

		TABLAS						
	No.	Nomb		No.	No.			
	Tabla	1101110	••	Atributos	Restricciones			
	, 1	DEPARTAM	ŒNTOS	2	1			
	2	CARRE	RAS	5	2			
CDEATE TABLE DEDARFMENTOC/				/				
CREATE TABLE DEPARTAMENTOS ( D# CHAR (2) NOT NULL,	- 1							
DNOMBRE CHAR(6) NOT NULL,	- 1				ATRIBU	ros		
DNOMBRE CHAR(0) NOT NOLL,		No.	l N	Vo.	Nombre	Tipo	Longit	ud No
CONSTRAINT PK DEPARTAMENTOS PRIMARY KEY	(D#)):	Tabla	Atr	ibuto		•		Nulo
		1		1	D#	CHAR	2	1
CREATE TABLE CARRERAS(		1		2	DNOMBRE	CHAR	6	1
C# CHAR(2) NOT NULL,		2		3	C#	CHAR	2	1
CNOMBRE CHAR(3) NOT NULL,	$\setminus$	2		4	CNOMBRE	CHAR	3	1
VIGENCIA CHAR(4) NOT NULL,	X	2		5	VIGENCIA	CHAR	4	1
SEMESTRES NUMERIC(2) NOT NULL,		2		6 S	EMESTRES	NUMERIC	2	1
D# CHAR(2) NOT NULL,		2		7	D#	CHAR	2	1
CONSTRAINT PK_CARRERAS PRIMARY KEY (C#) CONSTRAINT FK_CARRERAS FOREIGN KEY (D#)	N .	es departa	MENTOS					
				RESTR	ICCIONES			
	No.	No.	Tipo	N	ombre	Atributo	Tabla	Atributo
	Γabla Re	estricción				asociado		
	1	1	1	PK_DEPA	RTAMENTOS	1	-	-
	2	2	1	PK_C	ARRERAS	3	-	-
	2	3	2	FK C	ARRERAS	7	1	1

TABLAS							
No.	Nombre	No.	No.				
Tabla		Atributos	Restricciones				
1	DEPARTAMENTOS	2	1				
2	CARRERAS	5	2				

CREATE TABLE DEPARTAMENTOS (
D# CHAR (2) NOT NULL,
DNOMBRE CHAR (6) NOT NULL,

CONSTRAINT PK DEPARTAMENTOS PRIMARY KEY (D#));

CREATE TABLE CARRERAS()

C# CHAR(2) NOT NULL,
CNOMBRE CHAR(3) NOT NULL,
VIGENCIA CHAR(4) NOT NULL,
SEMESTRES NUMERIC(2) NOT NULL,
CHAR(4) NOT NULL,

D# CHAR(2) NOT NULL,

Nombre No. No. Tipo Longitud No Tabla Atributo Nulo D# CHAR 1 2 DNOMBRE CHAR 6 1 C# CHAR 2 4 CNOMBRE CHAR 3 1 VIGENCIA CHAR 5 4 1 SEMESTRES NUMERIC 2 7 D# CHAR 2 1

ATRIBUTOS

CONSTRAINT PK\_CARRERAS PRIMARY KEY (C#),
CONSTRAINT FK CARRERAS FOREIGN KEY (D#) REFERENCES DEPARTAMENTOS(D#));

	RESTRICCIONES									
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo				
Tabla	Restricción			asociado						
1	1	1	PK_DEPARTAMENTOS	1	-	-				
2	2	1	PK_CARRERAS	3	-	-				
2	3	2	FK CARRERAS	7	1	1				

			TA					
		No. Tabla	Nombre	No Atrib				
		1	DEPARTAMENTO	OS 2				
		, 2	CARRERAS	5	2			
CREATE TAB	LE DEPARTAMENTOS ( CHAR (2) NOT NULL,	<u></u>						
DNOMBRE	CHAR(6) NOT NULL,	\			ATRIBU	TOS		
DITOTIDICE	Similar industry		No.	No.	Nombre	Tipo	Longitud	No
CONSTRAINT	PK DEPARTAMENTOS PRIMARY KI	EY (D#)); \	Tabla A	Atributo				Nulo
	_		1	1	D#	CHAR	2	1
CREATE TAB	LE CARRERAS (		1	2	DNOMBRE	CHAR	6	1
C#	CHAR(2) NOT NULL		2	3	C#	CHAR	2	1
CNOMBRE	CHAR(3) NOT NULL,		2	4	CNOMBRE	CHAR	3	1
VIGENCIA	CHAR(4) NOT NULL,		2	5	VIGENCIA	CHAR	4	1
SEMESTRES	NUMERIC(2) NOT NULL,		2	6	SEMESTRES	NUMERIC	2	1
D#	CHAR(2) NOT NULL	L	_ 2	7	D#	CHAR	2	1
	PK_CARRERAS PRIMARY KEY (C: FK_CARRERAS FOREIGN KEY (D:		S DEPARTAMENT	OS (D#))	;			

	RESTRICCIONES									
No.	No.	Tipo	Nombre	Atributo	Tabla	Atributo				
Tabla	Restricción			asociado						
1	1	1	PK_DEPARTAMENTOS	1	-	-				
2	2	1	PK_CARRERAS	3	-	-				
2	3	2	FK CARRERAS	7	1	1				

			TAI	BLAS						
		No. Tabla	Nombre	N		No. Restricciones				
		1	DEPARTAMENTO	OS 2	2	1				
		, 2	CARRERAS		5	2				
CDEATE TARIE	E DEPARTAMENTOS (									
	CHAR(2) NOT NULL,	- 1								
	CHAR(2) NOT NULL,					ATRIBUT	os			
DNOMBRE	CHAR(0) NOT NOLL,		No.	No.		Nombre	Tipo	Long	gitud	No
CONSTRAINT I	PK DEPARTAMENTOS PRIMARY KEY	(T±1).	Tabla	tributo					·	Nulo
COMPTIMENT .		7",,,	1	1		D#	CHAR	- 2	2	1
CREATE TABLE	E CARRERAS (		1	2	Di	NOMBRE	CHAR	(	5	1
	CHAR(2) NOT NULL,	- 1	2	3		C#	CHAR		2	1
	CHAR(3) NOT NULL,	- 1	2	4	C	NOMBRE	CHAR	- 1	3	1
VIGENCIA	CHAR(4) NOT NULL,		2	5	V.	IGENCIA	CHAR	- 4	4	1
SEMESTRES	NUMERIC(2) NOT NULL,	1	2	6	SE	MESTRES	NUMERIC	: 2	2	1
D#	CHAR(2) NOT NULL,		2	7		D#	CHAR	- 2	2	1
	PK_CARRERAS PRIMARY KEY (C#) FK_CARRERAS FOREIGN KEY (D#)		S DEPARTAMENT							
				RE	STRIC	CCIONES				
		No. Γabla Res	No. Tipo stricción		Non	mbre	Atributo asociado	Tabla	Atrib	outo
		1	1 1	PK I	DEPAR	TAMENTOS	1	-/	-	
		2	2 1	P	K CA	RRERAS	3	-	-	
		2	3 2	F	K CA	RRERAS	7	1	1	
						·				

MSC. Martin Agúndez Amador 10

#### Gramática de atributos

```
CREATE TABLE DEPARTAMENTOS (
D#
            CHAR(2) NOT NULL,
            CHAR(6) NOT NULL
DNOMBRE
CONSTRAINT PK_DEPARTAMENTOS PRIMARY KEY (D#))
CREATE TABLE DEPARTAMENTOS (
      C#
                  CHAR (3) NOT NULL,
      CNOMBRE
      VIGENCIA
                  CHAR (4) NOT NULL,
      SEMESTRES
                  CHAR(2) NOT NULL,
                  CHAR(2) NOT NULL,
CONSTRAINT PK_CARRERAS PRIMARY KEY (C#),
CONSTRAINT FK CARRERAS FOREIGN KEY (D#)
                       REFERENCES DEPARTAMENTOS (D#))
```

No.	Reglas	Producciones
1	С	cti(A);B
2	В	C   I   λ
3	A	iD(d)EF
4	D	h   u
5	E	<u>e</u> g   λ
6	F	, G   λ
7	G	A   H
8	Н	biJ(i)K
9	J	p k   j k
10	K	$,H li(i)L \lambda$
11	L	, Η   λ
12	I	mqiv(M);N
13	M	OP
14	0	'a'   d
15	P	, M   λ
16	N	Ι С λ

## Gramática de atributos

No.	Reglas	Producciones
1	С	c t({Valida_NombreTabla}(A); B
2	В	C   I   λ
3	A	iD(d)EF
4	D	h   u
5	Е	e,g \lambda
6	F	, <b>G</b>  λ
	3	4 1 77

## Gramática de atributos

No.	Regla	Valor	Producciones
1	С	200	<b>16</b> 17 4 <b>700</b> 52 202 <b>3</b> 3 55 201
2	В	201	200   211   99
3	A	202	4/203 52 61 53 204 205
4	D	203	18   19
5	Е	204	20 21   99
6	F	205	50 206   99
	~/	226	0001007

No.	Reglas	Producciones
1	С	ti {Valida_NombreTabla} (A); B
2	В	$C   I   \lambda$
3	A	iD(d)EF
4	D	h   u
5	Е	<b>e</b> ,g   λ
6	F	, G   λ
_	~	A   TT

### Programación de la gramática de atributos

```
Inicio
   Insertar Pila ('$')
   Insertar Pila ('Q')
   Insertar '$' al final de Tabla Léxica
  APUN = Apuntador al primer Token de Tabla Léxica
                                                                 700
  Repetir
      X = Extraer_Pila()
                                                                  52
      K = Tabla Lexica[APUN]
      Si (X = TERMINAL) o (X = `\$') entonces
                                                                 202
          Si (X = K) entonces
            APUN++
                                                                  53
          Caso Contrario
                                                                  55
            ERROR()
      Caso Contrario
                                                                  201
          Si(M[X,K] = PRODUCCION) entonces
             Si (PRODUCCION <> '\lambda') entonces
                  Insertar Pila(PRODUCCION) ←
         Caso Contrario
            ERROR()
    Hasta X = 'S'
Fin
```

MSC. Martin Agúndez Amador 14

### Programación de la gramática de atributos

```
Inicio
  Insertar_Pila ('$')
                                          APUN = Apuntador al primer Token de Tabla Léxica
  Insertar_Pila ( 'Q')
  Insertar '5' al final de Tabla Léxica
                                          Repetir
  APUN = Apuntador al primer Token de Tabla Léxica
  Repetir
                                                K = Tabla Lexica[APUN]
     K = Tabla Lexica(APUN)
                                               X = Extraer Pila()
    X = Extraer_Pila()
    Si (X > = 700) entonces
                                               Si(X > = 700) entonces
        En Caso de X haz
          700: Si (Valida NombreTabla() = Verddo
                                                     En Caso de X haz
              ERROR()
                                                         700: Si (Valida NombreTabla() = Verdadero)
        Fin Caso
    Caso Contrario
                                                               ERROR()
       Si (X = TERMINAL) o (X = `$`) entonces
                                                    Fin Caso
          Si (X = K) entonces
            APUN++
                                               Caso Contrario
          Caso Contrario
            ERROR()
                                                    Si (X = TERMINAL) o (X = `\$') entonces
       Caso Contrario
          Si (M/X,K) = PRODUCCION) entonces
                                                         Si (X = K) entonces
             Si (PRODUCCION <> '\lambda') entonces
                                                           APUN++
                Insertar_Pila(PRODUCCION)
          Caso Contrario
                                                        Caso Contrario
            ERROR ()
                                                           ERROR ()
   Hasta X = 'S'
Fin
```

### Programación de la gramática de atributos

```
En Caso de X haz

700: S(Valida_NombreTabla()) Verdadero) entonces

ERROR()

Fin Caso

Valida_NombreTabla (NombreT: Char): Booleano

Inicio

AT ← 1

Valida_NombreTabla ← Falso

Mientras (AT <= NRegistros (Tablas) haz

Si (NombreT <> Tablas.NombreTabla[AT]) entonces

AT ++

Caso Contrario

Valida_NombreTabla ← Verdadero

AT ← NRegistros (Tablas)

Fin
```

