

## SINTAXIS DEL LENGUAJE SQL/92

En este anexo se va a presentar la sintaxis precisa del conjunto de instrucciones del lenguaje SQL/92 estudiadas. La notación que se va a seguir para presentar esta sintaxis es una versión extendida de BNF<sup>1</sup>. En BNF, cada elemento sintáctico se define mediante un conjunto de reglas de producción. Estas reglas definen un elemento en términos de una fórmula formada de caracteres, tiras de caracteres y elementos sintácticos que pueden ser usados para formar una instancia de dicho elemento. A continuación se incluyen los símbolos y reglas de la versión de BNF que se ha usado:

- *tira de caracteres en cursiva*: define un nombre de un identificador.
- tira de caracteres normal en minúsculas: define el nombre de un elemento sintáctico que no es un símbolo terminal y que se define en otra regla de producción.
- TIRA DE CARACTERES EN MAYÚSCULAS: es un símbolo terminal de la gramática y define una palabra reservada del lenguaje SQL.
- ::= El operador de producción. Se usa en una regla de producción para separar el elemento definido y la forma que lo define. El elemento que se define aparece a la izquierda del operador y la fórmula que lo define a la derecha.
- [ ] Los corchetes cuadrados sirven para indicar elementos opcionales dentro de una fórmula.
- { } Las llaves se usan para agrupar elementos en una fórmula. La porción de fórmula incluida entre llaves debe ser explícitamente especificada.
- | La barra vertical es el operador de alternativa que permite indicar que la porción de fórmula que sigue a la barra es una alternativa a la porción de la fórmula que le precede. Si la barra aparece en la fórmula sin estar incluida en unos corchetes o llaves indica una alternativa completa para los elementos definidos en la fórmula. Si aparece en una porción de la fórmula que está entre corchetes o llaves indica alternativas para el contenido del par de corchetes o llaves más internos.
- Si *xyz* es un elemento sintáctico, *lista\_xyz* es una lista de elementos del tipo *xyz* separados por algún tipo de separador (espacio en blanco, salto de línea, etc.).
- Si *xyz* es un elemento sintáctico, *comalista\_xyz* es una lista de elementos del tipo *xyz* separados por comas.

**Esta es la única documentación permitida en el examen de bases de datos y no debe llevar anotaciones.**

<sup>1</sup> Bakus Normal Form o Bakus Naur Form

## PRIMERA PARTE: SQL 92

### Definición de datos

- 1) definición\_esquema ::=  
 CREATE SCHEMA [*esquema*] [AUTHORIZATION *usuario*]  
 [lista\_elemento\_esquema]
- 2) elemento\_esquema ::=  

definición_dominio		definición_tabla		
definición_vista		definición_restricción		definición_privilegio
- 3) definición\_dominio ::=  
 CREATE DOMAIN [AS] tipo\_dato  
 [definición\_valor\_por\_defecto]  
 [lista\_definición\_restricción\_dominio]
- 4) definición\_valor\_por\_defecto ::=     DEFAULT {literal | función\_sistema | NULL }
- 5) definición\_restricción\_dominio ::=  
 [CONSTRAINT *restricción*] CHECK (expresión\_condicional) [cuando\_comprobar]
- 6) cuando\_comprobar ::=  
 [[NOT] DEFERRABLE] [INITIALLY {IMMEDIATE | DEFERRED}]
- 7) definición\_tabla ::=  
 CREATE TABLE *tabla\_base*  
 comalista\_definición\_columna[comalista\_definición\_restricción\_tabla]
- 8) definición\_columna ::=  
*columna* { tipo\_dato | dominio } [definición\_valor\_por\_defecto]  
 [lista\_definición\_restricción\_columna]
- 9) definición\_restricción\_columna ::=  
 [CONSTRAINT *restricción* ]  

{ NOT NULL		
PRIMARY KEY		
UNIQUE		
REFERENCES tabla [(comalista_		
columna)]		
[MATCH {FULL   PARTIAL } ]		
[ON DELETE acción_referencia]		
[ON UPDATE acción_referencia]		
CHECK (expresión_condicional) }		
[cuando_comprobar]		
- 10) acción\_referencia ::=  
 NO ACTION | CASCADE | SET DEFAULT | SET NULL

- 11) definición\_restricción\_tabla ::=  
 [CONSTRAINT *restricción* ]  
 {PRIMARY KEY |  
 UNIQUE (*comalista\_columna*) |  
 FOREIGN KEY (*comalista\_columna*) REFERENCES *tabla* [(*comalista\_columna*)]  
 [MATCH {FULL | PARTIAL } ]  
 [ON DELETE acción\_referencia]  
 [ON UPDATE acción\_referencia] |  
 CHECK (expresión\_condicional) }  
 [cuando\_comprobar]
- 12) definición\_vista ::=  
 CREATE VIEW *vista* [(*comalista\_columna*)]  
 AS expresión\_tabla [WITH CHECK OPTION]
- 13) definición\_restricción ::=  
 CREATE ASSERTION *restricción*  
 CHECK (expresión\_condicional) [cuando\_comprobar]
- 14) definición\_privilegio ::=  
 GRANT {*comalista\_privilegio* | ALL PRIVILEGES }  
 ON objeto TO {*comalista\_usuarios* | PUBLIC}  
 [WITH GRANT OPTION]
- 15) privilegio ::= SELECT | INSERT [(*comalista\_columna*)] |  
 UPDATE [(*comalista\_columna*)] | DELETE
- 16) objeto ::= DOMAIN *dominio* | [TABLE] *tabla*
- 17) alteración\_dominio ::=  
 ALTER DOMAIN *dominio*  
 {SET DEFAULT {literal | función\_sistema | NULL } |  
 DROP DEFAULT |  
 ADD definición\_restricción\_dominio |  
 DROP CONSTRAINT *restricción* }
- 18) alteración\_tabla ::=  
 ALTER TABLE *tabla\_base*  
 {ADD [COLUMN] *definición\_columna* |  
 ALTER [COLUMN] *columna*  
 {SET DEFAULT {literal | función\_sistema | NULL } |  
 DROP DEFAULT |  
 DROP [COLUMN] *columna* {RESTRICT | CASCADE} }
- 19) eliminación\_esquema ::= DROP SCHEMA *esquema* {RESTRICT | CASCADE}
- 20) eliminación\_dominio ::= DROP DOMAIN *dominio* {RESTRICT | CASCADE}

- 21) eliminación\_tabla\_base ::= DROP TABLE *tabla\_base* {RESTRICT | CASCADE}
- 22) eliminación\_vista ::= DROP VIEW *vista* {RESTRICT | CASCADE}
- 23) eliminación\_restricción\_general ::= DROP ASSERTION *restricción*
- 24) eliminación\_autorización ::=  
 REVOKE [GRANT OPTION FOR]  
 {ALL | SELECT | INSERT[(*comalista\_columna*)] |  
 DELETE | UPDATE [(*comalista\_columna*)]}  
 ON objeto TO {*comalista\_usuarios* | PUBLIC }  
 {RESTRICT | CASCADE}

## Expresiones de tabla

- 25) expresión\_tabla ::=  
 expresión\_concatenación\_tabla | expresión\_no\_concatenación\_tabla
- 26) expresión\_concatenación\_tabla ::=  
 | referencia\_tabla [NATURAL] [tipo\_concatenación] JOIN referencia\_tabla  
 [ON expresión\_condicional | USING (*comalista\_columna*) ]  
 | (expresión\_tabla) CROSS JOIN referencia\_tabla  
 | (expresión\_concatenación\_tabla)
- 27) referencia\_tabla ::=  
 tabla [[AS] *variable\_recorrido*] |  
 (expresión\_tabla) [[AS] *variable\_recorrido*] |  
 expresión\_concatenación\_tabla
- 28) tipo\_concatenación ::=  
 INNER | LEFT [OUTER] | RIGHT [OUTER] |  
 FULL [OUTER] | UNION
- 29) expresión\_conjuntista\_tabla ::=  
 término\_conjuntista\_tabla |  
 expresión\_tabla {UNION | EXCEPT } [ALL]  
 [CORRESPONDING [BY (*comalista\_columna*)]] término\_tabla
- 30) término\_conjuntista\_tabla ::=  
 tabla\_primaria\_conjuntista |  
 término\_tabla INTERSECT [ALL]  
 [CORRESPONDING [BY (*comalista\_columna*)]] tabla\_primaria
- 31) término\_tabla ::=  
 término\_conjuntista\_tabla | expresión\_concatenación\_tabla
- 32) tabla\_primaria ::=  
 tabla\_primaria\_conjuntista | expresión\_concatenación\_tabla

- 33) *tabla\_primaria\_conjuntista* ::=
- |                    |  |                               |  |
|--------------------|--|-------------------------------|--|
| TABLE <i>tabla</i> |  | constructor_tabla             |  |
| expresión_SELECT   |  | (expresión_conjuntista_tabla) |  |
- 34) *constructor\_tabla* ::= VALUES *comalista\_constructor\_fila*
- 35) *constructor\_fila* ::= expresión\_escalar | (*comalista\_expresión\_escalar*) | (*expresión\_tabla*)
- 36) *expresión\_SELECT* ::=
- |   |
|---|
| SELECT [ALL   DISTINCT] <i>comalista_item_seleccionado</i>                            |
| FROM <i>comalista_referencia_tabla</i>  |
| [WHERE <i>expresión_condicional</i> ]   |
| [GROUP BY <i>comalista_referencia_columna</i> [HAVING <i>expresión_condicional</i> ]] |
- 37) *item\_seleccionado* ::=
- |                                       |  |                                 |
|---------------------------------------|--|---------------------------------|
| expresión_escalar [AS] <i>columna</i> |  | [ <i>variable_recorrido</i> .]* |
|---------------------------------------|--|---------------------------------|

### Actualización de datos

- 38) *inserción* ::=
- |   |
|---|
| INSERT INTO <i>tabla</i> { [( <i>comalista_columna</i> )] <i>expresión_tabla</i>   DEFAULT VALUES } |
|---|
- 39) *actualización* ::=
- |   |
|---|
| UPDATE <i>tabla</i> SET <i>comalista_asignación</i> |
| [WHERE <i>expresión_condicional</i> ]               |
- 40) *asignación* ::= *columna* = { *expresión\_escalar* | DEFAULT | NULL }
- 41) *eliminación* ::= DELETE FROM *tabla* [WHERE *expresión\_condicional*]

### Expresiones condicionales

- 42) *expresión\_condicional* ::=
- |                     |  |  |
|---------------------|--|--|
| término_condicional |  | <i>expresión_condicional</i> OR <i>término_condicional</i> |
|---------------------|--|--|
- 43) *término\_condicional* ::=
- |                    |  |  |
|--------------------|--|--|
| factor_condicional |  | <i>término_condicional</i> AND <i>factor_condicional</i> |
|--------------------|--|--|
- 44) *factor\_condicional* ::= [NOT] *comprobación\_condicional*
- 45) *comprobación\_condicional* ::= *condición\_primaria* [IS [NOT] {TRUE | FALSE} ]
- 46) *condición\_primaria* ::= *condición\_simple* | (*expresión\_condicional*)
- 47) *condición\_simple* ::=
- |                              |  |                          |  |                       |  |                     |  |
|------------------------------|--|--------------------------|--|-----------------------|--|---------------------|--|
| <i>condición_comparación</i> |  | <i>condición_between</i> |  | <i>condición_like</i> |  | <i>condición_in</i> |  |
|------------------------------|--|--------------------------|--|-----------------------|--|---------------------|--|

- |                          |  |                        |  |                          |  |                         |  |                         |
|--------------------------|--|------------------------|--|--------------------------|--|-------------------------|--|-------------------------|
| <i>comprobación_nulo</i> |  | <i>condición_match</i> |  | <i>condición_all_any</i> |  | <i>condición_exists</i> |  | <i>condición_unique</i> |
|--------------------------|--|------------------------|--|--------------------------|--|-------------------------|--|-------------------------|
- 48) *condición\_comparación* ::= *constructor\_fila* *predicado\_comparación* *constructor\_fila*
- 49) *predicado\_comparación* ::= = | < | <= | > | >= | <>
- 50) *condición\_between* ::= *constructor\_fila* [NOT] BETWEEN *constructor\_fila*  
AND *constructor\_fila*
- 51) *condición\_like* ::= *expresión\_tira\_caracteres* [NOT] LIKE *patrón* [ESCAPE *escape*]
- 52) *condición\_in* ::=
- |  |  |
|--|--|
| <i>constructor_file</i> [NOT] IN ( <i>expresión_tabla</i> )              |  |
| <i>expresión_escalar</i> [NOT] IN ( <i>comalista_expresión_escalar</i> ) |  |
- 53) *compración\_nulo* ::= *constructor\_fila* IS [NOT] NULL
- 54) *condición\_match* ::=
- |  |
|--|
| <i>constructor_fila</i> MATCH [UNIQUE] [PARTIAL   FULL] ( <i>expresión_tabla</i> ) |
|--|
- 55) *condición\_all\_any* ::=
- |   |
|---|
| <i>constructor_fila</i> <i>predicado_comparación</i> {ALL   ANY   SOME}( <i>expresión_tabla</i> ) |
|---|
- 56) *condición\_exists* ::= EXISTS (*expresión\_tabla*)
- 57) *condición\_unique* ::= UNIQUE (*expresión\_tabla*)

### Expresiones escalares

- 58) *expresión\_escalar* ::=
- |                                  |  |
|----------------------------------|--|
| <i>expresión_numérica</i>        |  |
| <i>expresión_tira_caracteres</i> |  |
- 59) *expresión\_numérica* ::= *término\_numérico* |  
*expresión\_numérica* {+ | -} *término\_numérico*
- 60) *término\_numérico* ::= *factor\_numérico* |  
*término\_numérico* { \* | / } *factor\_numérico*
- 61) *factor\_numérico* ::= [+ | -] *numero\_primario*
- 62) *numero\_primario* ::=
- |                                   |  |                                    |  |
|-----------------------------------|--|------------------------------------|--|
| <i>referencia_columna</i>         |  | <i>literal</i>                     |  |
| <i>referencia_función_escalar</i> |  | <i>referencia_función_agregada</i> |  |
| ( <i>expresión_tabla</i> )        |  | ( <i>expresión_numérica</i> )      |  |
- 63) *referencia\_función\_agregada* ::= COUNT(\*)  
| {AVG | MAX | MIN | SUM | COUNT } ([ALL | DISTINCT] *expresión\_escalar*)
- 64) *expresión\_tira\_caracteres* ::= *concatenación\_tira\_caracteres* |  
*tira\_caracteres\_primaria*

65) concatenación\_tira\_caracteres ::=  
expresión\_tira\_caracteres ||<sup>2</sup> tira\_caracteres\_primaria

66) tira\_caracteres\_primaria ::=  
referencia\_columna | literal |  
función\_usuario | referencia\_función\_escalár |  
referencia\_función\_agregada | (expresión\_tabla) |  
(expresión\_tira\_caracteres)

### Miscelánea

67) tabla ::= *tabla\_base* | *vista*

68) patrón ::= expresión\_tira\_caracteres

69) escape ::= expresión\_tira\_caracteres

70) función\_sistema ::= función\_usuario | función\_tiempo

71) función\_usuario ::=  
USER |  
CURRENT\_USER |  
SESSION\_USER |  
SYSTEM\_USER

72) función\_tiempo ::=  
CURRENT\_DATE |  
CURRENT\_TIME |  
CURRENT\_TIMESTAMP

<sup>2</sup> “||” es el operador concatenación de tiras de caracteres

Para facilitar el manejo de la sintaxis, a continuación se listan todas las categorías sintácticas de SQL92 que se han definido ordenadas alfabéticamente indicando, con el número que las acompaña, en qué orden se puede encontrar su definición.

acción_referencia	10)
actualización	39)
alteración_dominio	17)
alteración_tabla	18)
asignación	40)
compración_nulo	53)
comprobación_condicional	45)
cuando_comprobar	6)
concatenación_tira_caracteres	65)
condición_all_any	55)
condición_between	50)
condición_comparación	48)
condición_exists	56)
condición_in	52)
condición_like	51)
condición_match	54)
condición_primaria	46)
condición_simple	47)
condición_unique	57)
constructor_fila	35)
constructor_tabla	34)
definición_columna	8)
definición_dominio	3)
definición_esquema	1)
definición_privilegio	14)
definición_restricción	13)
definición_restricción_columna	9)
definición_restricción_dominio	5)
definición_restricción_tabla	11)
definición_tabla	7)
definición_valor_por_defecto	4)
definición_vista	12)
elemento_esquema	2)
eliminación	41)
eliminación_autorización	24)
eliminación_dominio	20)
eliminación_esquema	19)
eliminación_restricción_general	23)
eliminación_tabla_base	21)
eliminación_vista	22)

escape	69)
expresión_concatenación_tabla	26)
expresión_condicional	42)
expresión_conjuntista_tabla	29)
expresión_escalar	58)
expresión_numérica	59)
expresión_SELECT	36)
expresión_tabla	25)
expresión_tira_caracteres	64)
factor_condicional	44)
factor_numérico	61)
función_sistema	70)
función_tiempo	72)
función_usuario	71)
inserción	38)
item_seleccionado	37)
numero_primario	62)
objeto	16)
patrón	68)
predicado_comparación	49)
privilegio	15)
referencia_función_agregada	63)
referencia_tabla	27)
tabla	67)
tabla_primaria	32)
tabla_primaria_conjuntista	33)
término_condicional	43)
término_conjuntista_tabla	30)
término_numérico	60)
término_tabla	31)
tipo_concatenación	28)
tira_caracteres_primaria	66)

## SEGUNDA PARTE: SQL DE ORACLE8

A continuación se presenta la sintaxis de aquellas instrucciones de Oracle 8 que no coinciden con SQL/92:

**Concatenación externa** en ORACLE8:

La sentencia en SQL/92:

```
SELECT [ALL | DISTINCT] comalista_item_seleccionado
FROM t1 LEFT OUTER JOIN t2 ON t1.A op t2.B AND .....
```

es equivalente en ORACLE8 a:

```
SELECT [ALL | DISTINCT] comalista_item_seleccionado
FROM t1, t2
WHERE t1.A op t2.B (+) AND ...
```

### Definición de relación básica

- Definición\_relación\_básica ::= CREATE TABLE nom\_relación  
(comalista\_elemento\_relación\_básica)
- elemento\_relación\_básica ::= definición\_atributo  
| restricción\_relación
- definición\_atributo ::= nom\_atributo tipo\_datos  
[DEFAULT (expresión)]  
[lista\_restricción\_atributo]
- tipo\_datos ::= | CHAR (longitud)  
| VARCHAR (longitud)  
| NUMBER [(precisión[, escala])]  
| DATE
- restricción\_atributo ::= [CONSTRAINT nombre\_restricción]  
{[NOT] NULL  
| UNIQUE  
| PRIMARY KEY  
| REFERENCES nom\_relación\* [(nom\_atributo\*)]  
[ON DELETE CASCADE]  
| CHECK (condición) }  
[cuando\_comprobar]
- restricción\_relación ::=  
[CONSTRAINT nombre\_restricción]  
{ UNIQUE (comalista\_nom\_atributo)  
| PRIMARY KEY (comalista\_nom\_atributo)  
| FOREIGN KEY (comalista\_nom\_atributo)  
REFERENCES nom\_relación\* [(comalista\_nom\_atributo\*)]  
[ON DELETE CASCADE]  
| CHECK (condición)}  
[cuando\_comprobar]
- cuando\_comprobar ::=  
[[NOT] DEFERRABLE] [INITIALLY {IMMEDIATE | DEFERRED}]

- 8) modificación\_relación ::= ALTER TABLE nombre\_relación  
     {ADD (*comalista\_elemento\_relación\_básica*)  
     | MODIFY (*comalista\_definición\_atributo*)  
     | DROP  
     | [VALIDATE | NOVALIDATE] ENABLE  
     | DISABLE } (*restricción*) }
- 9) restricción ::= {PRIMARY [CASCADE]  
     | UNIQUE (*comalista\_nombre\_atributo*) [CASCADE]  
     | CONSTRAINT nombre\_restricción }
- 10) definición\_vista ::= CREATE [OR REPLACE] VIEW *nombre\_vista*  
     [(*comalista\_nombre\_atributo*)] AS *sentencia\_SELECT*  
     [WITH CHECK OPTION]
- 11) definición\_operación\_grant ::= GRANT *comalista\_privilegios\_sistema*  
     TO {PUBLIC | *comalista\_usuario*}  
     [WITH ADMIN OPTION]
- 12) definición\_regla :=  
 {CREATE | REPLACE} TRIGGER nombre\_regla  
     {BEFORE | AFTER | INSTEAD OF} evento [*disyunción\_eventos*]  
     ON {nombre\_relación | nombre\_vista}  
     [ [REFERENCING OLD AS nombre\_referencia [NEW AS nombre\_referencia] ] ]  
     [FOR EACH {ROW | STATEMENT} {WHEN ( *condición* ) } ]  
     *bloque PL/SQL*
- 13) disyunción\_eventos := OR evento [*disyunción\_eventos*]
- 14) evento := INSERT | DELETE | UPDATE [OF *comalista\_nombre\_atributo*]

## Anexo: Lenguaje PL/SQL

### Estructura de un bloque PL/SQL:

DECLARE Sección de declaración de variables;

BEGIN Sentencias del bloque; END

### Sección de declaración de variables:

*nombre\_variable* tipo\_dato

tipo\_dato ::= {NUMBER | CHAR( ) | DATE }

### Sentencias del bloque PL/SQL:

secuencia\_de\_sentencias ::= sentencia [;secuencia\_de\_sentencias]

IF condición THEN secuencia\_de\_sentencias

[; ELSE secuencia\_de\_sentencias]; END IF

WHILE condición LOOP secuencia\_de\_sentencias ; END LOOP

FOR contador IN mínimo .. máximo LOOP secuencia\_de\_sentencias ; END LOOP

- Asignación: *nombre\_variable* := *expresión*

- Sentencias SQL: INSERT, DELETE, UPDATE, SELECT... INTO...

Manejo de errores: RAISE\_APPLICATION\_ERROR (*nro\_error*, '*mensaje*' )

Sentencias de entrada-salida: dbms\_output.put\_line ('*mensaje*').

Para facilitar el manejo de la sintaxis, a continuación se listan todas las categorías sintácticas de Oracle 8 que se han definido ordenadas alfabéticamente indicando, con el número que las acompaña, en qué orden se puede encontrar su definición.

cuando\_comprobar 7

definición\_atributo 3

definición\_operación\_grant 11

definición\_regla 12

definición\_relación\_básica 1

definición\_vista 10

disyunción\_eventos 13

elemento\_relación\_básica 2

evento 14

modificación\_relación 8

restricción 9

restricción\_atributo 5

restricción\_relación 6

tipo\_datos 4