

SINTAXIS DEL LENGUAJE SQL

En este anexo se va a presentar la sintaxis precisa del conjunto de instrucciones del lenguaje SQL estudiadas. La notación que se va a seguir para presentar esta sintaxis es una versión extendida de BNF¹. 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.

¹ Bakus Normal Form o Bakus Naur Form

PRIMERA PARTE:

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 dominio [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 (*comalista_columna*) |
 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}

Manipulación de datos

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 *tabla* | 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] comalista_item_seleccionado
 FROM comalista_referencia_tabla
 [WHERE expresión_condicional]
 [GROUP BY *comalista_columna* [HAVING expresión_condicional]]
- 37) item_seleccionado ::=
 expresión_escalar [AS] *columna* | [variable_recorrido.]*

Actualización de datos

- 38) inserción ::=
 INSERT INTO *tabla* { [(*comalista_columna*)] expresión_tabla | DEFAULT VALUES }
- 39) actualización ::=
 UPDATE *tabla* SET comalista_asignación
 [WHERE expresión_condicional]
- 40) asignación ::= *columna* = {expresión_escalar | DEFAULT | NULL }
- 41) eliminación ::= DELETE FROM *tabla* [WHERE expresión_condicional]

Expresiones condicionales

- 42) expresión_condicional ::=
 término_condicional | expresión_condicional OR término_condicional
- 43) término_condicional ::=
 factor_condicional | término_condicional AND factor_condicional
- 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 ::=
 condición_comparación | condición_between | condición_like | condición_in |
 comprobación_nulo | condición_match | condición_all_any | condición_exists |
 condición_único
- 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 ::=
 constructor_fila [NOT] IN (expresión_tabla) |
 expresión_escalares [NOT] IN (comalista_expresión_escalares)
- 53) comprobación_nulo ::= constructor_fila IS [NOT] NULL
- 54) condición_match ::=
 constructor_fila MATCH [UNIQUE] [PARTIAL | FULL] (expresión_tabla)
- 55) condición_all_any ::=
 constructor_fila predicado_comparación {ALL | ANY | SOME}(expresión_tabla)
- 56) condición_exists ::= EXISTS (expresión_tabla)
- 57) condición_único ::= UNIQUE (expresión_tabla)

Expresiones escalares

- 58) expresión_escalares ::= expresión_numérica |
 expresión_tira_caracteres |
- 59) expresión_numérica ::= término_numérico |
 expresión_numérica {+ | -} término_numérico
- 60) término_numérico ::= factor_numérico |
 término_numérico {* | /} factor_numérico
- 61) factor_numérico ::= [+ | -] número_primario
- 62) número_primario ::= referencia_columna | *literal* |
 referencia_función_escalares | referencia_función_agregada |
 (expresión_tabla) | (expresión_numérica)
- 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 ||² tira_caracteres_primaria

66) tira_caracteres_primaria ::=
referencia_columna | *literal* |
función_usuario | referencia_función_escalar |
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

² “||” 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_escalár	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 ESTÁNDAR DE ORACLE

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

Definición de relación básica

- 1) Definición_relación_básica ::= CREATE TABLE *nom_relación*
(comalista_elemento_relación_básica)
- 2) elemento_relación_básica ::= definición_atributo
| restricción_relación
- 3) definición_atributo ::= *nom_atributo* tipo_datos
[DEFAULT (expresión)]
[lista_restricción_atributo]
- 4) tipo_datos ::= | CHAR (longitud)
| VARCHAR (longitud)
| NUMBER [(precisión[, escala])]
| DATE
- 5) restricción_atributo ::= [CONSTRAINT *nombre_restricción*]
{ [NOT] NULL
| UNIQUE
| PRIMARY KEY
| REFERENCES *nom_relación* [(*nom_atributo*)]
[ON DELETE CASCADE]
| CHECK (expresión_condicional) }
[cuando_comprobar]
- 6) 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 (expresión_condicional) }
[cuando_comprobar]
- 7) cuando_comprobar ::=
[[NOT] DEFERRABLE] [INITIALLY {INMEDIATE | 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 expresión_SELECT
 [WITH CHECK OPTION]
- 11) definición_operación_grant ::= GRANT *comalista_privilegio*
 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 [WHEN (expresión_condicional)]]
 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