

## Resumen de Sintaxis

Fuentes: *"Introducción a Oracle 9i: SQL", Oracle Corporation.*  
*"Oracle 9i SQL Reference", Oracle Corporation.*

### Recuperación de datos: SELECT

#### Una tabla. Alias de columna.

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}
FROM      tabla;
Operadores aritméticos: +, -, *, /
Operador de concatenación: ||
```

#### Restricción de filas

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}
FROM      tabla
[WHERE     condición(es)];
Operadores de comparación: =, >, >=, <, <=, <>, !=
Otros: BETWEEN ... AND ..., IN, LIKE, IS NULL
Condiciones lógicas: NOT, AND, OR
```

#### Ordenamiento

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}
FROM      tabla
[WHERE     condición(es)]
[ORDER BY  {columna|expresión|alias} [ASC|DESC]];
```

#### Funciones de una fila

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}
FROM      tabla
[WHERE     condición(es)]
[ORDER BY  {columna|expresión|alias} [ASC|DESC]];
```

Funciones de caracteres: lower, upper, initcap, concat, substr, length, instr, lpad, rpad, trim  
 Funciones numéricas: round, trunc, mod, sign, ceil, floor  
 Funciones de fecha: sysdate, add\_months, last\_day, round, trunc  
 Funciones de conversión: to\_number, to\_date, to\_char  
 Funciones generales: nvl, nvl2, nullif, coalesce  
 Expresiones condicionales: case, decode

#### Varias tablas. Joins.

##### INNER JOIN

```
SELECT t1.columna, t2.columna
FROM   tabla1 t1, tabla2 t2, ...
[WHERE t1.columna = t2.columna]
[AND   condición(es)];
```

Join entre n tablas: **mínimo n-1 condiciones de join**  
 Join de desigualdad: BETWEEN ... AND ..., >=, <=

##### OUTER JOIN

```
SELECT t1.columna, t2.columna
FROM   tabla1 t1, tabla2 t2, ...
[WHERE t1.columna (+) = t2.columna];
SELECT t1.columna, t2.columna
FROM   tabla1 t1, tabla2 t2, ...
[WHERE t1.columna = t2.columna (+)];
```

**ANSI SQL 1999**

```

SELECT t1.columna, t2.columna
FROM   tabla1 t1
[CROSS JOIN tabla2] |
[NATURAL JOIN tabla2] |
[JOIN tabla2 USING (nombre_columna)] |
[JOIN tabla2
  ON (tabla1.nombre_columna = tabla2.nombre_columna)] |
[LEFT | RIGHT | FULL OUTER JOIN tabla2
  ON (tabla1.nombre_columna = tabla2.nombre_columna)];

```

**Funciones de varias filas**

```

SELECT funcion_de_grupo(columna), ...
FROM   tabla
[WHERE condición(es)];

```

Funciones de grupo: avg, count, max, min, sum

**Agrupamiento de datos**

```

SELECT columna, funcion_de_grupo(columna), ...
FROM   tabla
[WHERE condición(es)]
[GROUP BY expresión_de_agrupamiento]
[ORDER BY columna];

```

**Restricción de grupos**

```

SELECT columna, funcion_de_grupo(columna), ...
FROM   tabla
[WHERE condición(es)]
[GROUP BY expresión_de_agrupamiento]
[HAVING condición_de_grupo]
[ORDER BY columna];

```

**Extensiones al GROUP BY**

|  |   |
|--|---|
| <b>Subtotales</b><br><pre> SELECT columna, func_de_grupo(columna), ... FROM   tabla [WHERE condición(es)] [GROUP BY [ROLLUP] expr_de_agrupamiento] [HAVING condición_de_grupo] [ORDER BY columna]; </pre>  | <b>Tabulación cruzada</b><br><pre> SELECT columna, func_de_grupo(columna), ... FROM   tabla [WHERE condición(es)] [GROUP BY [CUBE] expr_de_agrupamiento] [HAVING condición_de_grupo] [ORDER BY columna]; </pre>                           |
| <b>Identificación de valores nulos</b><br><pre> SELECT columna, func_de_grupo(columna),   GROUPING(expresión), ... FROM   tabla [WHERE condición(es)] [GROUP BY [ROLLUP] [CUBE]   expr_de_agrupamiento] [HAVING condición_de_grupo] [ORDER BY columna]; </pre> | <b>Conjuntos de agrupamientos</b><br><pre> SELECT columna, func_de_grupo(columna), ... FROM   tabla [WHERE condición(es)] [GROUP BY [GROUPING SETS]   (conjuntos_de_agrupamiento)] [HAVING condición_de_grupo] [ORDER BY columna]; </pre> |

**Subconsultas - Anidamiento**

|   |   |  |
|---|---|--|
| <b>En cláusula WHERE</b><br><pre> SELECT lista_select FROM   tabla1 WHERE  expresión operador       (SELECT lista_select         FROM   tabla2); </pre> | <b>En cláusula HAVING</b><br><pre> SELECT lista_select FROM   tabla1 GROUP BY   expresión_de_agrupamiento HAVING expresión operador       (SELECT lista_select         FROM   tabla2); </pre> | <b>En cláusula FROM</b><br><pre> SELECT lista_select FROM   (SELECT lista_select         FROM   tabla) [WHERE condición(es)]; </pre> |
|---|---|--|

**Subconsulta correlacionada**

```

SELECT lista_select
FROM   tabla1 principal
WHERE  columna1 operador
      (SELECT lista_select
       FROM   tabla2 secundaria
       WHERE  expresión1 =
            principal expresión2);

```

Operadores de comparación de una sola fila: =, >, >=, <, <=, <>

Operadores de comparación de una varias filas: IN, ANY, ALL

Operador [NOT] EXISTS

**Cláusula WITH**

```

WITH
nombre_bloque1 AS
(con consulta 1)
[, nombre_bloque2 AS
(con consulta 2) [, ... ]]
SELECT   expresión
FROM     nombre_bloque1 [, nombre_bloque2]
        [tabla1]
[WHERE   condición(es)]
[GROUP BY expresión_de_agrupamiento]
[HAVING  condición_de_grupo]
[ORDER BY columna];

```

**Operadores de conjunto**

|  |                                      |                                  |
|--|--------------------------------------|----------------------------------|
| consulta1<br>UNION / UNION ALL<br>consulta2; | consulta1<br>INTERSECT<br>consulta2; | consulta1<br>MINUS<br>consulta2; |
|--|--------------------------------------|----------------------------------|

**DML****Inserción**

```

INSERT INTO tabla [(columna [, columna...])]
VALUES          (valor [, valor...]);

```

Esta sentencia inserta una fila cada vez.

```

INSERT INTO tabla [(columna [, columna...])]
subconsulta;

```

Esta sentencia inserta de una vez todas las filas que devuelve la subconsulta.

**Actualización**

```

UPDATE tabla
SET   columna = valor [, columna = valor, ...]
WHERE condición(es);

```

Esta sentencia actualiza de una vez todas las filas que cumplen las condiciones de la cláusula WHERE.

```

UPDATE tabla
SET   columna = (SELECT columna
                  FROM   tabla
                  [WHERE  condición])
[, columna = (SELECT columna
               FROM   tabla
               [WHERE  condición])]
[WHERE condición(es)];

```

Esta sentencia actualiza asignando valores obtenidos en una subconsulta.

---

**Fusión**

---

```

MERGE INTO tabla alias
  USING (tabla|vista|subconsulta) alias
  ON (condición de join)
  WHEN MATCHED THEN
    UPDATE SET
      columna1 = valor1
      [, columna2 = valor2 ...]
  WHEN NOT MATCHED THEN
    INSERT (lista de columnas)
    VALUES (lista de valores);

```

Esta sentencia verifica en la tabla de cláusula INTO la existencia de los datos que se encuentran en la tabla de cláusula USING, en base a la condición de cláusula ON. Si los datos existen, se ejecuta el update de la cláusula WHEN MATCHED THEN. Si no existen, se ejecuta el insert de la cláusula WHEN NOT MATCHED THEN.

---

**Borrado**

---

```

DELETE [FROM] tabla
[WHERE condición(es)];

```

Esta sentencia elimina de una vez todas las filas que cumplen las condiciones de la cláusula WHERE.

```

DELETE FROM tabla
WHERE      columna = (SELECT columna
                       FROM   tabla
                       [WHERE condición])
[AND      condición(es)];

```

Esta sentencia elimina las filas que cumplen la condición, seleccionando de otra tabla.

---

**Manejo de transacciones**

---

```

COMMIT;
ROLLBACK;
SAVEPOINT nombre;
ROLLBACK TO SAVEPOINT nombre;

```

Una transacción se valida al emitir una sentencia DDL o confirmarla explícitamente (COMMIT).

---

**SQL\*Plus**

---

---

**Conexión / Desconexión**

---

```

sqlplus [usuario [/password [@base]]]
EXIT

```

---

**Descripción de tablas**

---

```

DESC[RIBE] nombre_tabla|nombre_vista|nombre_sinonimo

```

---

**Personalización del Entorno**

---

```

SET variable valor
SHOW variable

```

|  |   |
|--|---|
| <b>COLSEP { _   texto }</b>            | Define el carácter separador entre columnas.  |
| <b>ECHO { ON   OFF }</b>               | Controla si la ejecución de comandos lista cada comando a medida que se ejecutan.                             |
| <b>FEED[BACK] { 6   n   OFF   ON }</b> | Define que se muestre el nro. de registros devueltos cuando se selecciona por lo menos ese nro. de registros. |
| <b>HEA[DING] { OFF   ON }</b>          | Controla si se imprimen los títulos de las columnas.  |
| <b>LIN[ESIZE] { 80   n }</b>           | Define el nro. de caracteres que se muestran en cada línea.   |

|   |  |
|---|--|
| <b>LONG { 80   n }</b>                      | Define el ancho máximo en bytes para mostrar valores de tipo CLOB, LONG, etc.  |
| <b>NULL texto</b>                           | Define el texto que representa un valor nulo en el resultado de un comando SELECT.   |
| <b>PAGES[IZE] { 24   n }</b>                | Define el nro. de líneas de cada página.   |
| <b>SERVEROUT[PUT] { ON   OFF } [SIZE n]</b> | Controla si debe visualizar la salida de los stored procedures o bloques PL/SQL. SIZE define el nro. de bytes de la salida que se pueden almacenar temporalmente. El valor por defecto es 2.000. n debe estar entre 2.000 y 1.000.000. |
| <b>TERM[OUT] { ON   OFF }</b>               | Controla la visualización de la salida de comandos ejecutados desde un script a pantalla.  |
| <b>TI[ME] { ON   OFF }</b>                  | Controla la visualización de la hora actual.   |
| <b>TIME[NG] { ON   OFF }</b>                | Controla la visualización de estadísticas de tiempo.   |
| <b>TRIM[OUT] { ON   OFF }</b>               | Determina si SQL*Plus elimina blancos al final de cada línea mostrada. No afecta a la salida a archivos.   |
| <b>WRA[P] { ON   OFF }</b>                  | Controla si debe alinear la visualización de una fila a la siguiente línea, si es demasiado larga para el ancho actual de línea. OFF trunca la fila.   |

---

### Edición del buffer

---

L[IST]  
L[IST] n  
L[IST] m n  
R[UN]

---

### Manipulación de archivos

---

S[AVE] nombre\_archivo[.extensión] [REP[LACE] | APP[END]]  
GET nombre\_archivo[.extensión]  
STA[RT] nombre\_archivo [.extensión]  
@nombre\_archivo[.extensión]  
ED[IT] [nombre\_archivo[.extensión]]  
SPO[OL] [nombre\_archivo[.extensión] | OFF]

---

### Interacción

---

&variable  
&&variable  
DEFINE  
UNDEFINE  
SET VERIFY {ON | OFF}

---

### Ejecución de archivos de comandos

---

SET variable valor

|   |  |
|---|--|
| <b>ARRAY[SIZE] { 20   n }</b>                   | Número de filas que SQL*Plus recupera a la vez.  |
| <b>DEF[INE] { &amp;   carácter   ON   OFF }</b> | Define el carácter utilizado para prefijar las variables de sustitución.   |
| <b>TRIMS[POOL] { ON   OFF }</b>                 | Determina si SQL*Plus elimina blancos al final de cada línea escrita en archivo. No afecta a la salida por pantalla. |

SHO[W] { variable | opción }

|                            |  |
|----------------------------|--|
| <b>ALL</b>                 | Lista la configuración de todas las opciones de SHOW.                      |
| <b>ERR[ORS]</b>            | Muestra errores de compilación (PL/SQL).                                   |
| <b>PARAMETERS [nombre]</b> | Muestra el valor actual para uno o todos los parámetros de inicialización. |
| <b>REL[EASE]</b>           | Muestra el nro. de release de Oracle.                                      |
| <b>SPOO[L]</b>             | Indica si la salida se está redireccionando.                               |
| <b>SQLCODE</b>             | Muestra el código de retorno de SQL de la última operación                 |
| <b>USER</b>                | Muestra el nombre de usuario conectado.                                    |

PRO[MPT] [ texto ]

**DDL****Creación de tablas**

Nomenclatura de tablas: **de 1 a 30 caracteres, comenzando por letra, conteniendo a-z A-Z 0-9 - \$ y #, no usando palabras reservadas ni nombres de objetos del mismo esquema.**

```
CREATE TABLE [esquema.]tabla
    (columna tipo_de_dato [DEFAULT expresion] [, ...]);
```

Crea una tabla con la estructura especificada.

```
CREATE TABLE [esquema.]tabla
    [(columna [, columna...])]
```

```
AS subconsulta;
```

Crea una tabla basada en la subconsulta, insertándole las filas devueltas.

```
DESC[RIBE] tabla;
```

**Modificación**

```
ALTER TABLE tabla
ADD          (columna tipo_de_dato [DEFAULT expresion]
             [, columna tipo_de_dato] ...);
```

Agrega una o más columnas, al final de la tabla.

```
ALTER TABLE tabla
MODIFY      (columna tipo_de_dato [DEFAULT expresion]
             [, columna tipo_de_dato] ...);
```

Modifica tipo de dato, tamaño, o valor por defecto de un o más columnas.

```
ALTER TABLE tabla
DROP        (columna);
```

Elimina una columna (sólo una a la vez) de la tabla.

**Borrado. Truncamiento.**

```
DROP TABLE tabla;
```

Suprime la tabla, borrando todos sus datos, en forma permanente.

```
RENAME nombre_viejo TO nombre_nuevo;
```

Renombra la tabla.

```
TRUNCATE TABLE tabla;
```

Elimina todas las filas de la tabla, en forma permanente.

**Creación de restricciones**

```
CREATE TABLE [esquema.]tabla
    (columna tipo_de_dato [DEFAULT expresion]
    [restriccion_a_nivel_columna],
    ...
    [restriccion_a_nivel_tabla][,...]);
```

Creación de restricción al crear la tabla.

```
ALTER TABLE [esquema.]tabla
ADD          [CONSTRAINT nombre_restriccion] tipo (columna);
```

Agregado de restricción sobre tabla existente.

```
ALTER TABLE [esquema.]tabla
DROP        PRIMARY KEY | UNIQUE (columna) |
            CONSTRAINT nombre_restriccion [CASCADE];
```

Borrado de restricción existente.

```
[restriccion_a_nivel_columna]:
columna [CONSTRAINT nombre_restriccion] tipo_restriccion
columna NOT NULL
columna UNIQUE
columna PRIMARY KEY
columna REFERENCES tabla(columna)
columna CHECK (condición)
```

```
[restriccion_a_nivel_tabla]:
columna,...
[CONSTRAINT nombre_restriccion] tipo_restriccion (columna,...),
UNIQUE (columna)
PRIMARY KEY (columna [, columna...])
FOREIGN KEY (columna) REFERENCES tabla(columna)
CHECK (condición)
```

---

### Creación de sinónimos

```
CREATE [PUBLIC] SYNONYM sinonimo
FOR objeto;
```

---

### Borrado

```
DROP [PUBLIC] SYNONYM sinonimo;
```

---

### Creación de índices

```
CREATE INDEX indice
ON tabla (columna [, columna]...);
```

---

### Borrado

```
DROP INDEX indice;
```

---

### Reconstrucción

```
ALTER INDEX indice REBUILD;
```

---

### Creación de vistas

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW vista
[(alias [, alias]...)]
AS subconsulta
[WITH CHECK OPTION [CONSTRAINT restriccion]]
[WITH READ ONLY [CONSTRAINT restriccion]];
```

---

### Borrado

```
DROP VIEW vista;
```

---

### Creación de secuencias

```
CREATE SEQUENCE secuencia
[INCREMENT BY n]
[START WITH n]
[{MAXVALUE n | NOMAXVALUE}]
[{MINVALUE n | NOMINVALUE}]
[{CYCLE | NOCYCLE}]
[{CACHE n | NOCACHE}];
```

---

### Modificación

```
ALTER SEQUENCE secuencia
[INCREMENT BY n]
[{MAXVALUE n | NOMAXVALUE}]
[{MINVALUE n | NOMINVALUE}]
[{CYCLE | NOCYCLE}]
[{CACHE n | NOCACHE}];
```

---

**Borrado**

```
DROP SEQUENCE secuencia;
```

---

**Creación de DBLinks**

```
CREATE [PUBLIC] DATABASE LINK dblink  
[CONNECT TO usuario IDENTIFIED BY password]  
USING 'nombre_servicio';
```

---

**Borrado**

```
DROP [PUBLIC] DATABASE LINK dblink;
```

---

**Creación de Vistas Materializadas**

```
CREATE MATERIALIZED VIEW vista_materializada  
[ON PREBUILT TABLE]  
{NEVER REFRESH | [REFRESH {FAST|COMPLETE|FORCE}  
  [ON {DEMAND|COMMIT}]]}  
[{START WITH|NEXT} fecha]  
AS subconsulta;
```

---

**Modificación / Refresco**

```
ALTER MATERIALIZED VIEW vista_materializada  
REFRESH {FAST|COMPLETE|FORCE}  
[ON {DEMAND|COMMIT}]  
[{START WITH|NEXT} fecha];
```

---

**Borrado**

```
DROP MATERIALIZED VIEW vista_materializada;
```

---

**Creación de Logs de Vistas Materializadas**

```
CREATE MATERIALIZED VIEW LOG ON tabla  
[WITH {OBJECT ID|PRIMARY KEY|ROWID|SEQUENCE}  
  (columna [, columna...])]  
[ {INCLUDING|EXCLUDING} NEW VALUES];
```

---