



 **ACADEMY**

Programación de Bases de Datos con SQL

19-3

Revisión del Examen Final



ORACLE ACADEMY

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

En esta lección se abordan los siguientes objetivos:

- Revisar los puntos clave sobre la manipulación de mayúsculas/minúsculas y caracteres
- Revisar las funciones generales y de número, fecha, conversión
- Revisar las expresiones condicionales
- Revisar el producto cartesiano y las operaciones de unión
- Revisar las uniones no igualitarias, uniones externas, autouniones, uniones cruzadas, uniones naturales y cláusulas de unión
- Revisar las funciones de grupo, la sintaxis group by y las cláusulas having

Objetivos

En esta lección se abordan los siguientes objetivos:

- Escribir subconsultas de una y varias filas
- Revisar las subconsultas pareadas y no pareadas
- Revisar las subconsultas correlacionadas
- Revisar las sentencias DML insert, update, delete, merge y las inserciones en varias tablas
- Revisar las sentencias DDL CREATE, ALTER, RENAME, TRUNCATE, FLASHBACK TABLE, DROP y FLASHBACK QUERY
- Revisar las sentencias DCL CREATE y los privilegios de objetos REVOKE

Objetivo

- La revisión es la mejor preparación para la evaluación.
- La evaluación le permite observar todo lo que ha aprendido y resalta las áreas en las que puede que desee mejorar.
- La revisión de los temas aprendidos hasta este punto le ayudará a ser el mejor durante el examen final.

Revisión de la Sintaxis

- Se trata de una revisión de la sintaxis.
- Asegúrese de que también revisa las reglas relativas a la sintaxis.
- Estas se tratan con profundidad a lo largo del curso.

Manipulación de Mayúsculas/Minúsculas y de Caracteres

- Mayúsculas/minúsculas

```
LOWER(column name|expression)
UPPER(column name|expression)
INITCAP(column name|expression)
```

- Caracteres

```
CONCAT(column name|expression, column name|expression)
SUBSTR(column name|expression,n,m)
LENGTH(column name|expression)

INSTR(column name|expression, string literal)
LPAD (column name|expression, n, character literal)
RPAD(column name|expression, n, character literal)
TRIM ( [leading | trailing | both] char1 FROM char2)
REPLACE (column name|expression, string to be replaced, replacement
string)
```

Funciones Numéricas

```
ROUND (column|expression,n)  
TRUNC (column|expression,n)  
MOD (column|expression, column|expression)
```

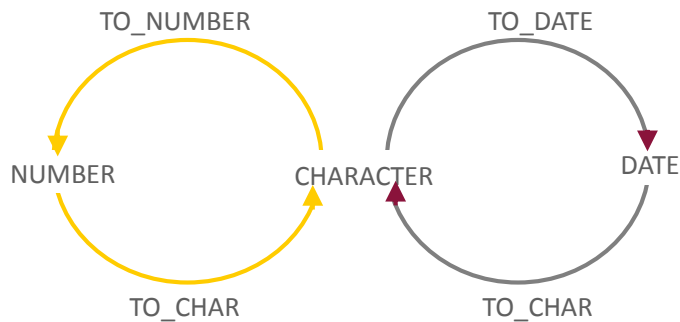

Funciones de Fecha

```
ROUND (column|expression,string)
TRUNC (column|expression,string)

MONTHS_BETWEEN (column|expression, column|expression)
ADD_MONTHS (column|expression,n)
NEXT_DAY (column|expression,'day')
LAST_DAY (column|expression)
```

Funciones de Conversión

```
TO_CHAR(number, 'format model')  
TO_CHAR(date, 'format model')  
TO_NUMBER(character string, 'format model')  
TO_DATE(character string, 'format model')
```



Funciones NULL

```
NVL(column|expression, value)
```

```
NVL2(column|expression, column|expression,  
      column|expression)
```

```
NULLIF(column|expression, column|expression)
```

```
COALESCE(column|expression, column|expression,  
          column|expression... column|expression)
```

Expresiones Condicionales

- Específico de Oracle

```
DECODE(column1|expression, search1, result1  
      [, search2, result2,...,]  
      [, default])
```

- ANSI

```
CASE expr WHEN comparison_expr1 THEN return_expr1  
      [WHEN comparison_expr2 THEN return_expr2  
      WHEN comparison_exprn THEN return_exprn  
      ELSE else_expr]  
END
```

Sintaxis Estándar ANSI SQL

- Cross Join

```
SELECT last_name, department_name  
FROM employees CROSS JOIN departments;
```

- Natural Join

```
SELECT employee_id, last_name, department_name  
FROM employees NATURAL JOIN departments;
```

- Join .. On

```
SELECT e.employee_id, e.last_name, e.salary, j.grade_level  
FROM employees e JOIN job_grades j  
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

Sintaxis Estándar ANSI SQL

- Joins .. Using

```
SELECT employee_id, last_name, department_name  
FROM employees JOIN departments  
USING (department_id);
```

- Join .. On

```
SELECT e.employee_id, e.last_name, d.department_id, d.location_id  
FROM employees e JOIN departments d  
ON (e.department_id = d.department_id);
```

Sintaxis Estándar ANSI SQL

Uniones Externas

- Right Outer Join

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e RIGHT OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

- Left Outer Join

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e LEFT OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```

Sintaxis Estándar ANSI SQL

Uniones Externas

- Full Outer Join (no es comparable con la unión específica de Oracle)

```
SELECT e.employee_id, e.last_name,  
       e.department_id, d.department_name  
FROM employees e FULL OUTER JOIN departments d  
ON (e.department_id = d.department_id);
```


Funciones de Grupo, Sintaxis de Group By y Cláusulas HAVING

```
AVG (column |expression)
COUNT (column |expression)
MIN (column |expression)
MAX (column |expression)
SUM (column |expression)
VARIANCE (column |expression)
STDDEV (column |expression)
```

```
SELECT column1, AVG (column |expression)
FROM table 1
GROUP BY (ROLLUP | CUBE) (column1 | GROUPING SETS)
HAVING AVG (column |expression)
```

Subconsultas de Una o Varias Filas

```
SELECT column1..  
FROM table 1  
WHERE column2 = (SELECT column2  
                  FROM table 1  
                  WHERE column 3 = expression)
```

- Operadores de Una Sola Fila: =, >, <, >=, <=, <>
- Operadores de Varias Filas: IN, ANY, ALL

Subconsultas Pareadas y No Pareadas

- Pareadas

```
SELECT column1..  
FROM table 1  
WHERE (column2, column3) = (SELECT column2, column3  
                             FROM table 1  
                             WHERE column 4 = expression);
```

- No pareadas

```
SELECT column1..  
FROM table 1  
WHERE column2 = (SELECT column2  
                 FROM table 1  
                 WHERE column 4 = expression)  
AND column3 = (SELECT column3  
               FROM table 2  
               WHERE column 4 = expression);
```

Subconsultas Correlacionadas

```
SELECT o.column1..  
FROM table_1 o  
WHERE o.column2 = (SELECT i.column2  
                   FROM table_2 i  
                   WHERE i.column1 = o.column1)
```

Inserción, Actualización y Supresión de Datos

- INSERT explícito

```
INSERT INTO table (column1, column2...)  
VALUES (value1, value2...) ;
```

- INSERT implícito

```
INSERT INTO table  
VALUES (value1, value2, value3, value4);
```

```
UPDATE table1  
SET column1 = value1,  
    column2 = value2...  
WHERE column1 = value;
```

```
DELETE FROM table1  
WHERE column1 = value;
```

Inserción, Actualización y Supresión de Datos

```
UPDATE table1  
SET column1 = value1,  
    column2 = value2...  
WHERE column1 = value;
```

```
DELETE FROM table1  
WHERE column1 = value;
```

Inserción, Actualización y Supresión de Datos

```
conditional_insert_clause  
[ ALL | FIRST ]  
WHEN condition THEN  
    insert_into_clause [ values_clause ]  
WHEN condition THEN  
    insert_into_clause [ values_clause ]  
ELSE insert_into_clause [ values_clause ]
```

Valores por Defecto

```
CREATE TABLE table1 (  
column1      DATE DEFAULT SYSDATE,...)  
  
INSERT INTO table1  
  (column1,...)  
VALUES  
  (DEFAULT,...) ;
```


Sentencia Merge

Inserción en varias tablas

```
MERGE INTO destination-table USING source-table  
ON matching-condition  
WHEN MATCHED THEN UPDATE  
SET .....  
WHEN NOT MATCHED THEN INSERT  
VALUES (.....) ;
```

Creación de Tablas

```
CREATE TABLE table  
(column data type [DEFAULT expression],  
column data type [DEFAULT expression],  
.....[ ] );
```

```
CREATE TABLE tablename  
[(column, column, ...)]  
AS subquery;
```

Especificación de Tipos de Dato

```
NUMBER (p, s)
CHAR
VARCHAR2 (n)
DATE
TIMESTAMP
TIMESTAMP WITH TIMEZONE
TIMESTAMP WITH LOCAL TIME ZONE
INTERVAL YEAR TO MONTH
INTERVAL DAY TO SECOND
CLOB
BLOB
RAW
```

Modificación de una Tabla

```
ALTER TABLE tablename  
ADD (column_name data type [DEFAULT expression]...);
```

```
ALTER TABLE tablename MODIFY (column_name VARCHAR2(30));
```

```
ALTER TABLE tablename DROP COLUMN column name;
```

```
ALTER TABLE tablename SET UNUSED (column name);
```

```
ALTER TABLE tablename DROP UNUSED COLUMNS;
```

Modificación de una Tabla

```
DROP TABLE tablename;
```

```
FLASHBACK TABLE tablename TO BEFORE DROP;
```

```
SELECT * FROM user_recyclebin;
```

```
SELECT versions_starttime "START_DATE",  
       versions_endtime "END_DATE",  
       column, column.....  
FROM table  
   VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE  
WHERE  column = value;
```

Restricciones a Nivel de Columna

```
CREATE TABLE table
(col1 data type CONSTRAINT tab_col1_pk PRIMARY KEY,
 col2 data type CONSTRAINT tab_col2_nn NOT NULL,
 col3 data type CONSTRAINT tab_col3_uk UNIQUE,
 col4 data type CONSTRAINT tab_col4_ck CHECK (col4 > value),
 col5 data type CONSTRAINT tab_col5 REFERENCES table2 (col1));
```

Restricciones a Nivel de Tabla

```
CREATE TABLE table
(col1 data type,
 col2 data type,
 col3 data type,
 col4 data type,
 col5 data type,
 CONSTRAINT tab_col1_pk PRIMARY(col1),
 CONSTRAINT tab_col3_uk UNIQUE(col2),
 CONSTRAINT tab_col4_ck CHECK (col4 > value),
 CONSTRAINT tab1_col5_fk FOREIGN KEY (col5) REFERENCES table2 (col1));
```

Creación y Gestión de Vistas

```
CREATE [OR REPLACE] [FORCE| NOFORCE] VIEW view [(alias [, alias]...)]  
AS subquery  
[WITH CHECK OPTION [CONSTRAINT constraint]]  
[WITH READ ONLY [CONSTRAINT constraint]];
```

```
DROP VIEW viewname;
```


Análisis de N Principales

```
SELECT ROWNUM as RANK, col1, col2
FROM (SELECT col1, col2 FROM table1
ORDER BY col1)
WHERE ROWNUM <= n;
```

Vistas en Línea

```
SELECT t1.col1, t2.col2...  
FROM table 1 t1, (SELECT col1, col2..  
    FROM table2  
    WHERE ...) t2  
WHERE .....;
```

Creación de Secuencias

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

```
DROP SEQUENCE sequence_name;
```

Creación de Índices y Sinónimos

```
CREATE INDEX index_name  
ON table_name( column...,column);
```

```
DROP INDEX index_name;
```

```
CREATE [PUBLIC] SYNONYM synonym  
FOR object;
```

```
DROP [PUBLIC] SYNONYM name_of_synonym
```

Creación y Revocación de Privilegios de Objeto

```
CREATE USER user  
IDENTIFIED BY password;
```

```
GRANT privilege [, privilege...]  
TO user [, user| role, PUBLIC...];
```

```
ALTER USER user  
IDENTIFIED BY password;
```

Creación y Revocación de Privilegios de Objeto

```
CREATE ROLE role_name;
```

```
GRANT object_priv [(column_list)]  
ON object_name  
TO {user|role|PUBLIC}  
[WITH GRANT OPTION];
```

```
REVOKE {privilege [, privilege...]|ALL}  
ON object  
FROM {user[, user...]|role|PUBLIC}  
[CASCADE CONSTRAINTS];
```

Resumen

En esta lección, debe haber revisado lo siguiente:

- Puntos clave sobre la manipulación de mayúsculas/minúsculas y caracteres
- Funciones generales y de número, fecha, conversión
- Expresiones condicionales
- Producto cartesiano y las operaciones de unión
- Uniones no igualitarias, uniones externas, autouniones, uniones cruzadas, uniones naturales y cláusulas de unión
- Funciones de grupo, sintaxis de GROUP BY y Cláusulas HAVING

Resumen

En esta lección, debe haber revisado lo siguiente:

- Subconsultas de una o varias filas
- Subconsultas pareadas y no pareadas
- Subconsultas correlacionadas
- Sentencias DML insert, update, delete, merge y las inserciones en varias tablas
- Sentencias DDL, FLASHBACK TABLE, DROP y FLASHBACK QUERY



 **ACADEMY**