

ESQUEMA SQL					
INTRODUCCIÓN					
ALTER SESSION set NLS_DATE_FORMAT = 'dd/mm/yyyy';			ALTER SESSION set NLS_LANGUAGE = 'spanish';		
@ C:\scripts BDEMP.sql;	Select * from nombre_tabla;	desc nombre_tabla;	select * from DICTIONARY;	select * from user_tables;	select table_name from user_tables ;
ADMINISTRACIÓN					
CREATE TABLESPACE nombre DATAFILE 'nombre.dbs' SIZE nº [M/K] [DEFAULT STORAGE (INITIAL tamaño NEXT tamaño, MINEXTENTS tamaño, MAXEXTENTS tamaño, PCINCREASE valor)] [ONLINE/OFFLINE];			ALTER TABLESPACE nombre {[ADD Datafile 'nombre_archivo' [SIZE Nº [K/M]] [AUTOEXTEND ON/OFF] } [RENAME Datafile 'archivo' to 'archivo'] [DEFAULT STORAGE clausulas almacen] [ONLINE/OFFLINE]];		
DROP TABLESPACE 'nombre' [INCLUDING CONTENTS];			SELECT TABLESPACE_NAME from user_tablespaces;		
CREATE USER 'nombre' IDENTIFIED BY password [DEFAULT TABLESPACE nombre] [TEMPORARY TABLESPACE nombre] [QUOTA nº[K/M] ON nombre_tablespace] ;			ALTER USER 'nombre' IDENTIFIED BY password [DEFAULT TABLESPACE nombre] [TEMPORARY TABLESPACE nombre] [QUOTA nº[K/M] ON nombre_tablespace] ;		
DROP USER usuario [CASCADE];			DESC all_users; SELECT USERNAME from all_users;		
Ver privilegios del usuario actual: SELECT * from session_privs;			Ver privilegios del sistema asignados al usuario: SELECT * from user_sys_privs;		
Conceder privilegios: GRANT {objeto1, [objeto2]...} ALL [PRIVILEGES]]{[Col1, [Col2]]} ON [USUARIO.]nombre_tabla to {usuario 20} [WITH GRANT OPTION];			Retirar privilegios: REVOKE {objeto1, [objeto2]...} ALL [PRIVILEGES]] ON [USUARIO.]nombre_tabla FROM {Usuario rol};		
CREATE ROLE nombre_rol [IDENTIFIED BY password];			DROP ROLE nombre_rol;		
CREATE SEQUENCE nombre de secuencia [INCREMENT BY entero] [START WITH entero] [MAXVALUE entero NOMAXVALUE] [MINVALUE entero NOMINVALUE] [CYCLE NOCYCLE] [ORDER NOORDER] [CACHE entero NOCACHE];			Valor actual de la sec: Nombresecuencia.CURRVAL; Sig. valor de la sec: Nombresecuencia.NEXTVAL;		
DDL					
CREATE TABLE tabla (campo1 tipo(tamaño) [not null primary key], campo2 tipo (tamaño)) [CONSTRAINT nombre_restriccion (PRIMARY KEY (campo)/UNIQUE KEY (campo)/CHECK (campo condicion)/NOT NULL (campo)/FOREIGN KEY (campo) references nombretabla (campotabla));]			CREATE TABLE nombre_tabla AS SELECT * FROM otra_tabla;	CREATE TABLE nombre_tabla (campo1, campo2) AS SELECT campo1tabla2, campo2tabla2 FROM tabla2;	
ALTER TABLE nombre_tabla [ADD (columna tipo(tamaño)), [MODIFY (columna nuevo_tipo_de_dato)], [ADD CONSTRAINT nombre_restriccion] [DROP CONSTRAINT nombre_restriccion] [DISABLE CONSTRAINT nombre_restriccion];				DROP TABLE [USUARIO].nombre_tabla [CASCADE CONSTRAINT];	
CREATE [OR REPLACE] VIEW nombre_vista (columna1, columna2) As Consulta;				DROP VIEW Nombre_vista;	
SELECT * from user_Constraints where table_name = 'nombretabla';			SELECT CONSTRAINT_NAME from user_Constraints where table_name = 'nombretabla';		
DML					
INSERT INTO nombre-tabla VALUES (valores: números, 'letras o palabras');		INSERT INTO nombre-tabla (columna1, columna2) VALUES (valor1,valor2);		INSERT INTO nombre-tabla (columna1,columna2) SELECT (sentencia Select);	
DELETE FROM nombre-tabla [WHERE condición];		UPDATE nombre-tabla SET columna1= "nuevovalor" [,columna2= valornuevo2] [WHERE <condición>;		UPDATE nombre-tabla SET (columna1 ,columna2)=(SELECT col1, col2 from otratabla where [condicion]) [WHERE condición];	
SELECT [ALL DISTINCT] [* / columna1, columna2] FROM nombre-tabla [(WHERE condición)/[GROUP BY columna1, columna2] [HAVING condición-selección-grupos/] [ORDER BY columna1[DESC],columna2[DESC]];					
SUBCONSULTAS, JOINS Y COMPLEJAS					
Subconsulta: SELECT <columna1>, <columna2> FROM <nombre-tabla1>, <nombre-tabla2> WHERE <columna1> = (SELECT <columna1> FROM <nombre-tabla1>, <nombre-tabla2> WHERE <condición>;					
Join de igualdad: SELECT <columna1, columna2> FROM <nombre-tabla1 alias1>, <nombre-tabla2 alias2>WHERE alias1.Columna1 = alias2.Columna1;					
Join externo: SELECT <col1, col2, col3> FROM tabla1 alias, tabla2 alias WHERE alias1.col1(+)=alias2.col1;					

Autounion: `SELECT Alias1.columna1, Alias2.columna2 FROM tabla alias1 , tabla alias2 WHERE alias1.Columna2 = alias2.Columna1(+);`

Complejas: `SELECT col1 From tabla1 Where EXISTS/NOT EXISTS (select * from tabla2 where tabla2.col2 = tabla1.col2 and tabla2.col3 = 'x');`

FUNCIONES

Aritmeticas:

ABS(n)= valor absoluto de (n).

CEIL(n)= valor entero inmediatamente superior o igual a "n".

FLOOR(n)= valor entero inmediatamente inferior o igual a "n".

MOD(m, n)= resto de dividir "m" entre "n".

NVL (valor, expresión)= Sustituye un valor nulo por otro valor.

POWER (m, exponente)= potencia de un número.

ROUND (numero [, m])= Redondea números con el número de dígitos de precisión indicados.

SIGN (valor)= Indica el signo del "valor".

SQRT(n)= Devuelve la raíz cuadrada de "n".

TRUNC (numero, [m])= Trunca números para que tengan una cierta cantidad de dígitos de precisión.

VARIANCE (valor)= Devuelve la varianza de un conjunto de valores.

De cadena de caracteres:

CHR(n)= Devuelve el carácter cuyo valor en binario es equivalente a "n".

CONCAT(cad1, cad2)= Devuelve "cad1" concatenada con "cad2".

LOWER(cad)= Devuelve "cad" en minúsculas.

UPPER(cad)= Devuelve "cad" en mayúsculas.

INITCAP(cad)=Convierte la cadena "cad" a tipo titulo.

LPAD(cad1, n[,cad2])= Añade caracteres a la izquierda de la cadena tantos caracteres como indica la n y el carácter indicado en cad2

RPAD (cad1, n[,cad2])=Añade caracteres a la derecha de la cadena hasta que tiene una cierta longitud.

LTRIM (cad [,set])=Suprime blancos a la izquierda.

RTRIM (cad [,set])=Suprime blancos a la derecha.

REPLACE (cad, cadena_búsqueda [, cadena_sustitucion])=Sustituye un carácter o caracteres de una cadena con 0 o más caracteres. Devuelve la cadena con cada ocurrencia de cadena de búsqueda sustituida por la cadena de sustitución.

SUBSTR (cad, m [,n])=Obtiene parte de una cadena. Desde la posición m, muestra n caracteres.

LENGTH (cad)= Devuelve el número de caracteres de cad.

Para el manejo de fechas:

SYSDATE= Devuelve la fecha del sistema.

ADD_MONTHS(fecha, n)= Devuelve "fecha" incrementada en "n" meses.

LASTDAY (fecha)= Devuelve la fecha del último día del mes que contiene "fecha".

MONTHS_BETWEEN(fecha1, fecha2)=Devuelve la diferencia en meses entre "fecha1" y "fecha2".

NEXT_DAY(fecha, cad)= Devuelve la fecha del primer día de la semana indicado por "cad" después de la fecha indicada por "fecha".

Funciones de conversión:

TO_CHAR(d, fmt)=Transforma una fecha "d" a tipo varchar2 en el formato fmt especificado.

TO_DATE(char,fmt)=Transforma una cadena de caracteres de tipo varchar2 a un tipo DATE en el formato elegido

TO_NUMBER(char,fmt)=Transforma una cadena de caracteres en NUMBER bajo el formato indicado

Funciones de grupos de valores:

AVG(n)= Calcula el valor medio de "n" ignorando valores nulos.

COUNT (* | Expresión)= Cuenta el número de veces que la expresión evalúa algún dato.

MAX (expresión)= Calcula el máximo.

MIN (expresión)= Calcula el mínimo.

SUM (expresión)= Obtiene la suma de los valores de la expresión.

GREATEST (valor1, valor2...)= Obtiene el mayor valor de la lista.

LEAST (valor1, valor2...)= Obtiene el menor valor de la lista.

Otras Funciones:

DECODE(VAR, VAL1,COD1,VAL2,COD2...,Valor por defecto): Si var es igual a cualquier valor de la lista devuelve el correspondiente código, en caso contrario se obtiene el valor señalado como valor por defecto.

GREATEST(expr1, expr2, ... expr_n): devuelve el mayor valor de una lista de expresiones.

FORMATOS

yyyy año sin signo, **yyy** últimos 3 dígitos del año, **yy** últimos 2 dígitos del año, **y** último dígitos del año.

ww Número de semana del año, **w** Número de semana del mes.

mm Número del mes.

ddd Número del día del año, **dd** Número del día del mes.

syear o year año en ingles letra, **month** nombre mes en letra, **mon** abreviatura de 3 letras del nombre del mes, **day** nombre del día de la semana, **dy** abreviatura de 3 letras del nombre del día.