

Resumen SQL

Tipo de Dato:

- **VARCHAR2**(tamaño) : Almacena cadena de caracteres de longitud variable, entre () el max.
- **CHAR**(tamaño) : Almacena caracteres con una longitud fija especificada.
- **NUMBER**(precisión, escala) : Almacena datos numéricos

Operadores:

- + : Suma
- - : Resta
- * : Multiplicación
- / : División
- = : Igual a
- > : Mayor que
- >= : Mayor o igual que
- < : Menor que
- <= : Menor o igual que
- != : Distinto
- <> : Distinto
- **MAX** : Devuelve el valor máximo de una serie de valores.
- **AND** : Devuelve el valor TRUE cuando las dos condiciones son verdaderas
- **OR** : Devuelve el valor TRUE cuando una de las condiciones es verdadera
- **NOT** : Devuelve el valor TRUE si la condición es falsa
- % : Comodín, representa cualquier cadena de 0 o mas caracteres.
- _ : Marcador de posición que representa a un carácter cualquiera.
- **LIKE** : Para comparación de cadenas de caracteres con inclusión de % y _ .
- **NULL** : Para saber si el valor de una columna es nulo se usa columna **IS NULL** y si se quiere saber si no es nulo, se usa **IS NOT NULL** .
- **IN** : Para comprobar si una expresión pertenece o no (NOT) a un conjunto de valores.
- **BETWEEN** : Comprueba si un valor esta dentro de un rango de valores.

Consulta de los Datos, Sentencias:

SELECT columna **FROM** tabla **WHERE** condición **ORDER BY** columna/constante/expresión/función;

- **DESC** : Describe, da una descripción de la Tabla, se usa seguido del nombre de la Tabla.
- **FROM** : Especifica la tabla o lista de tablas de las que se recuperan los datos.
- **WHERE** : Obtiene las filas que cumplen la condición.
- **ORDER BY** : Especifica el criterio de clasificación del resultado de la consulta.
- **DESC** : Cuando se usa con ORDER BY, ordena de forma descendente.
- **DISTINCT** : Aparecen solo las filas que son distintas.
- * : Representa todas las columnas de la tabla.
- "alias" : Se utiliza para dar otro nombre a la columna en la presentación.

DESC Alumnos ;
Descripción de la tabla Alumnos.

SELECT Nombre **FROM** Alumnos ;
Visualiza los datos de la columna 'Nombre' de la Tabla 'Alumnos'.

SELECT Nombre, Apellidos **FROM** Alumnos **WHERE** (Nota>=5) AND (Curso=1) ;
Visualiza las columnas 'Nombre' y 'Apellidos' de la Tabla 'Alumnos' con la condición Nota<=5 y Curso=1

SELECT Nombre, Apellidos **FROM** Alumnos **WHERE** Nombre='Pedro' ;
Visualiza las columnas 'Nombre' y 'Apellidos' de la Tabla 'Alumnos' con la condición Nombre = Pedro

SELECT Nombre, Apellidos **FROM** Alumnos **ORDER BY** Nombre **DESC** ;

Visualiza las columnas 'Nombre' y 'Apellidos' de la Tabla 'Alumnos' ordenada por el Nombre en descendente

SELECT DISTINCT Dept_N **FROM** Emple ;

Visualiza solo las filas distintas de la columna 'Dept_N' de la Tabla 'Emple'.

SELECT * FROM Alumnos ;

Selecciona todas las columnas de la Tabla Alumnos.

SELECT Dept_N "Departamento" **FROM** Emple ;

El nombre de la columna Dept_N sera visualizado como Departamento.

SELECT Apellido **FROM** Emple **WHERE** Apellido **LIKE** 'A%o%' ;

Obtiene aquellos apellidos que empiezan por una 'A' y contienen una 'o' en su interior.

SELECT Apellido **FROM** Emple **WHERE** Comision **IS NOT NULL** ;

Consulta de los Apellidos de Empleados cuya Comisión no sea nula.

SELECT Apellido **FROM** Emple **WHERE** Dept_N **IN** (10, 30) ;

Consulta de los Apellidos de Empleados cuyo departamento sea 10 o 30.

SELECT Apellido **FROM** Emple **WHERE** Salario **BETWEEN** 1500 AND 2000 ;

Consulta de los Apellidos de Empleados cuyo Salario este comprendido entre 1500 y 2000.

SELECT Nombre, (Nota1+Nota2+Nota3)/3 "Nota media" **FROM** Notas_Alumnos **WHERE** Notas2<6 ;

De una tabla de Notas, selecciona lo alumnos cuya Nota2 es menor que 6, presenta la columna del nombre y otra columna que llamamos Nota media (por medio del alias), con el resultado de la media de las columnas Nota1, Nota2 y Nota3.

Operadores en SubConsultas:

Estos se usan cuando se sabe que el resultado es único.

- = : Igual a
- > : Mayor que
- >= : Mayor o igual que
- < : Menor que
- <= : Menor o igual que
- != : Distinto
- <> : Distinto
- **IN** : Para comprobar si una expresión pertenece o no (NOT) a un conjunto de valores. Este se usa cuando no se sabe si el resultado es único o va a dar mas de un resultado.

SELECT APELLIDO, OFICIO, SALARIO, FECHA_ALT **FROM** EMPL **WHERE** OFICIO=(**SELECT** OFICIO **FROM** EMPL **WHERE** APELLIDO='JIMENEZ') **OR** SALARIO>=(**SELECT** SALARIO **FROM** EMPL **WHERE** APELLIDO='FERNANDEZ') ;

Subconsulta, Queremos sacar Apellido, Oficio,.....de la Tabla EMPL cuando tengan el mismo Oficio que Jimenez o Salario igual o superior al que tiene Fernandez, entonces tenemos que hacer una consulta sobre otras 2 consultas como se ve en las instrucciones.

SELECT OFICIO, APELLIDO **FROM** EMPL **WHERE** DEPT_NO=20 **AND** OFICIO **IN** (**SELECT** OFICIO **FROM** EMPL **WHERE** DEPT_NO=(**SELECT** DEPT_NO **FROM** DEPART **WHERE** DNOMBRE='VENTAS'));

En esta Subconsulta como devuelve mas de un valor se utiliza el operador **IN**.

Subconsultas correlacionadas:

El formato es el siguiente: **SELECT * FROM** EMPL **E WHERE** SALARIO = (**SELECT** MAX (SALARIO) **FROM** EMPL **WHERE** DETP_NO = **E**.DEPT_NO);

Obtener los datos de los empleados cuyo salario sea el máximo salario de su departamento, para referenciar a dicho departamento se necesita un alias (**E**).

Combinación de Tablas en una consulta:

El formato es el siguiente: **SELECT** Columna **FROM** Tabla1, Tabla2 **WHERE** Tabla1.Columna = Tabla2.Columna;

SELECT APENOM "Alumno", NOMBRE "Asignatura" **FROM** ALUMNOS, ASIGNATURAS, NOTAS **WHERE** ALUMNOS.DNI=NOTAS.DNI **AND** NOTAS.COD=ASIGNATURAS.COD **AND** NOMBRE='FOL';

Obtener los alumnos que estudien la asignatura de FOL, para poder hacer esta consulta hay que combinar las tablas de Alumnos, Notas y Asignaturas.

Funciones de valores simples:

- **ABS** (n) : Devuelve el valor absoluto de 'n', es siempre un numero positivo.
- **CEIL** (n) : Obtiene el valor entero inmediatamente superior o igual a 'n'.
- **FLOOR** (n) : Lo opuesto a CEIL, devuelve el valor entero inmediatamente inferior o igual a 'n'.
- **MOD** (m, n) : Devuelve el resto resultante de dividir 'm' entre 'n'.
- **NVL** (valor, expresión) : Si 'valor' es nulo lo sustituye por la 'expresión', si no lo es devuelve 'valor'.
- **POWER** (m, exponente) : Calcula la potencia de un numero, devuelve el valor de 'm' elevado a un 'exponente'.
- **ROUND** (numero [,m]) : Devuelve el valor de 'numero' redondeado a 'm' decimales, si se omite 'm' devuelve 'numero' con 0 decimales y redondeado.
- **SIGN** (valor) : Indica el signo del 'valor', si menor que 0 devuelve -1, si mayor que 0 devuelve 1.
- **SQRT** (n) : Devuelve la raíz cuadrada de 'n', el valor de 'n' no puede ser negativo.
- **TRUNC** (numero, [m]) : Devuelve 'numero' truncado a 'm' decimales, si 'm' es negativo trunca por la izquierda del punto decimal, si se omite 'm' devuelve 'numero' con 0 decimales.

SELECT ABS (-46) FROM DUAL;

Devuelve valor: **46**

SELECT CEIL (2.5) FROM DUAL;

Devuelve valor: **3**

SELECT FLOOR (2.5) FROM DUAL;

Devuelve valor: **2**

SELECT MOD (10,3) FROM DUAL;

Devuelve valor: **1**

SELECT SALARIO + NVL (COMISION, 0) FROM EMPLE;

SELECT POWER (3,2) FROM DUAL;

Devuelve valor: **9**

SELECT ROUND (-33.67,1) FROM DUAL;

Devuelve valor: **-33,7**

SELECT TRUNC (-33.67,1) FROM DUAL;

Devuelve valor: **-33,6**

SELECT SQRT (9) FROM DUAL;

Devuelve valor: **3**

Funciones de grupos de valores:

- **AVG**(n) : Calcula el valor medio de 'n' ignorando los valores nulos.
- **COUNT** (* | expresión) : Cuenta el numero de veces que la expresión evalúa algún dato con valor no nulo, la opción '*' cuenta todas las filas seleccionadas.
- **MAX** (expresión) : Calcula el máximo valor de la 'expresión'.
- **MIN** (expresión) : Calcula el mínimo valor de la 'expresión'.
- **SUM** (expresión) : Obtiene la suma de valores de la 'expresión' distintos de nulos.
- **VARIANCE** (expresión) : Obtiene la varianza de los valores de 'expresión' distintos de nulos.

Se puede usar DISTINCT y ALL en este tipo de funciones, aunque su uso mas normal es con COUNT, de la siguiente manera:
COUNT (* | [DISTINCT] expresión) o COUNT (* | [ALL] expresión)

SELECT AVG (SALARIO) FROM EMPLE WHERE DEPT_NO=10;

SELECT COUNT (COMISION) FROM EMPLE;

SELECT MAX (SALARIO) FROM EMPLE;

SELECT MIN (SALARIO) FROM EMPLE;

SELECT SUM (SALARIO) FROM EMPLE;

SELECT VARIANCE (SALARIO) FROM EMPLE;

SELECT COUNT (DISTINCT OFICIO) "Oficio" FROM EMPLE;

Funciones de listas:

- **GREATEST** (valor1, valor2, ...) : Obtiene el mayor valor de la lista.
- **LEAST** (valor1, valor2,) : Obtiene el menor valor de la lista.

SELECT NOMBRE_ALUMNO GREATEST (NOTA1, NOTA2, NOTA3) "Nota Mayor", LEAST (NOTA1, NOTA2, NOTA3) "Nota Menor" FROM DUAL;

Funciones de cadenas de caracteres:

- **CHR** (n) : Devuelve el carácter cuyo valor en binario es equivalente a 'n'.
- **CONCAT** (cad1, cad2) : Devuelve 'cad1' concatenada con 'cad2'. es equivalente al operador ||.
- **LOWER** (cad) : Devuelve la cadena 'cad' con todas sus letras convertidas a minúsculas.
- **UPPER** (cad) : Devuelve la cadena 'cad' con todas sus letras convertidas a mayúsculas.
- **INITCAP** (cad) : Convierte la cadena 'cad' a tipo titulo, la primera letra de cada palabra de 'cad' a mayúsculas y el resto a minúsculas.
- **LPAD** (cad1, n [, cad2]) : Devuelve 'cad1' con longitud 'n' y ajustado a la derecha, 'cad2' es la cadena con la que se rellena por la izquierda, si se suprime 'cad2' el carácter de relleno es blanco.
- **RPAD** (cad1, n [, cad2]) : Devuelve 'cad1' con longitud 'n' y ajustado a la izquierda, 'cad2' es la cadena con la que se rellena por la derecha, si se suprime 'cad2' el carácter de relleno es blanco.
- **LTRIM** (cad, n [, set]) : Devuelve 'cad' con el grupo de caracteres 'set' omitidos por la izquierda, si se omite el segundo parámetro devuelve la misma cadena y si tiene blancos los omite por la izquierda.
- **RTRIM** (cad, n [, set]) : Devuelve 'cad' con el grupo de caracteres 'set' omitidos por la derecha, si se omite el segundo parámetro devuelve la misma cadena y si tiene blancos los omite por la derecha.
- **SUBSTR** (cad, m [,n]) : Devuelve la subcadena de 'cad' que abarca desde la posición indicada en 'm' hasta tantos caracteres como indique el numero'n',
- **TRANSLATE** (cad1, cad2, cad3) : Devuelve 'cad1' con los caracteres encontrados en 'cad2' y sustituidos por los caracteres de 'cad3', los caracteres que no están en 'cad2' no se cambian.
- **REPLACE** (cad, cadena_búsqueda [,cadena_sustitución]) : Devuelve 'cad' con cada ocurrencia de 'cadena-búsqueda' sustituida por 'cadena_sustitución'.

SELECT CHR (75), CHR (65) FROM DUAL;

SELECT CONCAT (LOWER (APELLIDO) || ' es ', UPPER (OFICIO)) "Puestos Empleados" FROM EMPLE;

SELECT CONCAT (INITCAP (APELLIDO) || ' es ', INITCAP (OFICIO)) "Puestos Empleados" FROM EMPLE;

SELECT LPAD (NOMBRE, 30, ' . ') "Izquierda", RPAD (NOMBRE, 30, ' . ') "Derecha" FROM EMPLE;

SELECT RTRIM (TUTULO, ' * ') FROM MISTEXTOS;

SELECT LTRIM (TUTULO, ' * ') FROM MISTEXTOS;

SELECT APELLIDO "Apellido", SUBSTR (APELLIDO, 1, 1) "Inicial Apellido" FROM EMPLE;

SELECT TRANSLATE ('OGRO', 'O', 'AS') FROM DUAL;

SELECT REPLACE (APELLIDO, 'EZ', 'O') "Apellido empleados" FROM EMPLE;

Funciones que devuelven valores numéricos:

- **ASCII** (cad) : Devuelve el valor ASCII de la primera letra de la cadena 'cad'.
- **LENGTH** (cad) : Devuelve el numero de caracteres de 'cad'.
- **INSTR** (cad1, cad2 [, comienzo [, m]]) : Devuelve la posición de la 'm_esima' ocurrencia de 'cad2' en 'cad1', empezando la búsqueda en la posición 'comienzo'. Por omisión, empieza buscando en la posición 1.

SELECT ASCII ('A') FROM DUAL;

SELECT TEMA, LENGTH (TEMA) "Longitud Tema" FROM LIBRERIA;

SELECT INSTR (AUTOR, (',')) "Apellido" FROM LIBROS;

Funciones para el manejo de fechas:

- **SYSDATE** : Devuelve la fecha del sistema.
- **ADD_MONTHS** (fecha, n) : Devuelve la fecha 'fecha' incrementada en 'n' meses.
- **LAST_DAY** (fecha) : Devuelve la fecha del ultimo día del mes que contiene 'fecha'.
- **NEXT_DAY** (fecha, cad) : Devuelve la fecha del primer día de la semana indicado por 'cad' Después de la fecha indicada por 'fecha', el día de la semana se indica por su nombre, lunes (monday).
- **MONTHS_BETWEEN** (fecha1, fecha2) : Devuelve la diferencia en meses entre las fechas 'fecha1' y 'fecha2'.

SELECT SYSDATE FROM DUAL;

SELECT FECHA_ALT, ADD_MONTHS (FECHA_ALT, 12) FROM EMPLE WHERE DEPT_NO=10;

SELECT FECHA_ALT, LAST_DAY (FECHA_ALT) FROM EMPLE WHERE DEPT_NO=10;

SELECT TRUNC (MONTHS_BETWEEN (SYSDATE, '18/11/1964') / 12) "Edad actual" FROM DUAL;

Funciones de conversión:

- **TO_CHAR** (fecha, 'formato') : Convierte una 'fecha' (de tipo DATE) a tipo VARCHAR2 en el 'formato' especificado.
- **TO_CHAR** (numero, 'formato') : Convierte una 'numero' (de tipo NUMBER) a tipo VARCHAR2 en el 'formato' especificado.
- **TO_DATE** (cad, 'formato') : Convierte 'cad' de tipo VARCHAR2 o CHAR, a un valor tipo DATE según el 'formato' especificado.
- **TO_NUMBER** (cadena [, 'formato']) : Convierte la 'cadena' a tipo NUMBER según el 'formato' especificado.

SELECT NOMBRE, FECHANAC, TO_CHAR (FECHANAC, '"Nacio el " dd " de " month " de " yyyy') "Fecha Formateada" FROM NACIMIENTOS;

SELECT TO_NUMBER ('123.456', '999G999') "No Convierte" FROM DUAL;
En este caso no convierte al tener el numero ya marcado el separador de miles.

SELECT TO_DATE ('01012006') FROM DUAL;

Mascaras de formato de caracteres

- **year** año en ingles.
- **month** Nombre del mes (ENERO).
- **mon** Abreviatura de 3 letras del nombre del mes (ENE).
- **day** Nombre del día de la semana (LUNES).
- **a.m/p.m.** Muestra a.m. o p.m. dependiendo del momento del dia.
- **b.c./a.d.** Indicador para el año (antes de cristo o después de cristo).

Mascaras de formato numéricas

- **cc** Valor del siglo.
- **yyyy** Año sin signo.
- **yyy** Últimos 3 dígitos del año, si solo se ponen 2 son 2 dígitos y con una y 1 dígito.
- **q** Numero del trimestre.
- **ww** Numero de semana del año.
- **w** Numero de semana del mes.
- **mm** Numero de mes.
- **ddd** Numero del día del año.
- **dd** Numero del día del mes.
- **d** Numero del día de la semana.
- **hh** Hora (1-12).
- **hh24** Hora (1-24).
- **mi** Minutos.
- **ss** Segundos.
- **ssss** segundos transcurridos desde medianoche.
- **j** Juliano

Tablas de formatos numéricos

- **9** Devuelve el valor con el numero especificado de dígitos, ejemplo 999 para 3 dígitos.
- **\$** Devuelve el valor con el signo dolar a la izquierda, ejemplo \$9999.
- **D** Devuelve el carácter decimal en la posición especificada, ejemplo 9D99, seria 1,23.
- **G** Devuelve el carácter de miles en la posición indicada, ejemplo 9G999, seria 1.000.
- **L** Devuelve el símbolo de la moneda local en la posición indicada, ejemplo 999L, 100€.
- **,** Devuelve la coma en la posición especificada, ejemplo 9,999, seria 1,234.
- **.** Devuelve el punto en la posición especificada, ejemplo 9.999, seria 1.000.
- y otros tantos mas.....

Otras Funciones:

- **DECODE** (var, val1, cod1, val2, cod2 ..., valor_por defecto): Esta función sustituye un valor por otro, si "var" es igual a cualquier valor de la lista ("val1", "val2"..) devuelve el correspondiente código ("cod1", "cod2"...), en caso contrario da el valor señalado por defecto.
- **GROUP BY** Consulta de datos según grupos determinados, por ejemplo salario medio de cada departamento.
- **HAVING** Es una condición al "GROUP BY" para controlar cual de los conjuntos de filas se visualiza.
- **COUNT (*)** Usada con GROUP BY realiza el conteo de las filas determinadas por este.

SELECT APELLIDO, OFICIO, **DECODE** (**UPPER** (OFICIO), 'PRESIDENTE', 1, 'EMPLEADO', 2, 5) "Código" **FROM** EMPLE;

De la tabla EMPLE selecciona todas las filas y codifica el OFICIO, si el OFICIO es PRESIDENTE codifica con un 1, si es EMPLEADO con un 2, si es otra cosa con un 5.

SELECT DEPT_NO, OFICIO, **COUNT**(*) **FROM** EMPLE **GROUP BY** DEPT_NO, OFICIO;

Visualiza a partir de la tabla EMPLE el numero de empleados que hay en cada departamento.

Combinación externa (OUTER JOIN)

Es una variedad de combinación de tablas, que permite seleccionar algunas filas de una tabla aunque estas no tengan correspondencia con las filas de la otra tabla con la que se combina:

- **SELECT** tabla1.column1, tabla1.column2, tabla2.column1
FROM tabla1, tabla2
WHERE tabla1.column1 = tabla2.column1(+);
- Se seleccionan todas las filas de la tabla1, aunque no tengan correspondencia con las filas de la tabla2; se denota con el símbolo (+) detrás de la columna de la tabla2 (que es donde no se encuentran las filas)

SELECT D.DEPT_NO, DNOMBRE, **COUNT**(E.EMP_NO) **FROM** EMPLE E, DEPART D **WHERE** E.DEPT_NO (+) = D.DEPT_NO **GROUP BY** D.DEPT_NO, DNOMBRE;

Se muestran todos los departamentos aunque no tengan empleados, gracias a añadir (+), si no se añade este símbolo solo se muestran los departamentos que tienen empleados (el 40 que no tiene empleados no aparece).

Operadores UNION, INTERSECT y MINUS:

- **UNION** Combina los resultados de 2 consultas, no muestra filas duplicadas.
- **UNION ALL** Combina los resultados de 2 consultas y muestra las filas duplicadas.
- **INTERSECT** Realiza la intersección, devuelve las filas que son iguales en las 2 consultas, no muestra filas duplicadas.
- **MINUS** Hace la resta, devuelve las filas que están en la 1ª Select, pero no están en la segunda.

SELECT NOMBRE **FROM** ALUM **UNION** **SELECT** NOMBRE **FROM** NUEVOS;
Combina las tablas ALUM y NUEVOS y muestra la columna NOMBRE

SELECT NOMBRE **FROM** NUEVOS **INTERSECT** **SELECT** NOMBRE **FROM** ANTIGUOS;
Realiza la intersección de las Tablas NUEVOS y ANTIGUOS

SELECT NOMBRE **FROM** ALUM **MINUS** (**SELECT** NOMBRE **FROM** NUEVOS **UNION** **SELECT** NOMBRE **FROM** ANTIGUOS);
Hace la Unión de las tablas NUEVOS y ANTIGUOS y el resultado lo resta de la tabla ALUM

Manipulación de Datos INSERT, UPDATE, DELETE:

- **INSERT INTO** Tabla (columna1, columna2, ...) **VALUES** (valor1, valor2, ..)
Se utiliza para insertar datos en las tablas (valor1 va con columna1). Si el valor es alfanumérico va entre ' '.
- **INSERT INTO** Tabla (columna1, columna2, ...) **SELECT** (columna1, columna2,...) **FROM** Tabla (clausulas de SELECT)
Inserción de datos mediante una los datos obtenidos con SELECT.
- **UPDATE** Tabla **SET** columna=valor **WHERE** condición
Se utiliza para modificar datos en las tablas. Si el valor es alfanumérico va entre ' '.
- **UPDATE** Tabla **SET** columna=(clausulas de SELECT) **WHERE** condición
Modificar datos mediante los datos obtenidos con SELECT. Si el valor es alfanumérico va entre ' '.
- **DELETE FROM** Tabla **WHERE** condición
Eliminación de una o varias filas de una tabla. Condición necesaria para especificar que filas se borran. La condición se puede realizar con una SELECT.

INSERT INTO PROFESORES (COD_CENTRO, DNI, APELLIDOS, ESPECIALIDAD)
VALUES (22, 23444800, 'Gonzalez Sevilla, Miguel A.', 'HISTORIA');

INSERT INTO ALUM **SELECT** * **FROM** NUEVOS **MINUS** **SELECT** * **FROM** ALUM;
Inserta en la Tabla ALUM los alumnos nuevos obtenidos de la selección, en este caso las dos tablas tienen las mismas columnas.

UPDATE EMPLE **SET** SALARIO=SALARIO+100, COMISION=COMISION+10 **WHERE** DEPT_NO=10;
Sube en 100€ el salario y en 10€ la comisión de los empleados del departamento 10

UPDATE CENTROS **SET** (DIRECCION, NUM_PLAZAS) = (**SELECT** DIRECCION, NUM_PLAZAS **FROM** CENTROS **WHERE** COD_CENTROS = 50) **WHERE** COD_CENTRO = 10;
Esta sentencia actualiza los datos de Dirección y plazas del centro con código 10 con los que hay en el centro con código 50.

DELETE FROM PROFESORES WHERE DNI IN (SELECT DNI FROM PROFESORES MINUS SELECT DNI FROM PERSONAL);

Borra a los profesores que estén en la tabla PROFESORES y que no estén en la tabla PERSONAL.

Manipulación de Datos ROLLBACK, COMMIT Y AUTOCOMMIT:

- **COMMIT** Se usa para validar los datos introducidos.
- **AUTOCOMMIT** Se usa para validar automáticamente las transacciones.
SHOW AUTOCOMMIT Se usa para saber en que modo esta AUTOCOMMIT.
SET AUTOCOMMIT Se usa para activar o desactivar el AUTOCOMMIT con ON y OFF
- **ROLLBACK** Esta orden aborta las transacciones volviendo la base de datos al estado del ultimo COMMIT.

Creación, Supresión y Modificación de tablas:

- **CREATE TABLE** Tabla Crea objetos de base de datos. Tablas, vistas, sinónimos, etc.
(
Columna1 Tipo_dato [NOT NULL], Define columna, tipo dato y si es NO NULL (no puede tener valores nulos)
Columna2 Tipo_dato [DEFAULT], con DEFAULT define que ponga un valor por omisión.
.....
CONSTRAINT nombre **PRIMARY KEY** (Columna1, Columna2,...), Define la/las columnas que son clave primaria.
CONSTRAINT nombre **FOREIGN KEY** (Columna1, Columna2,...), Define la clave ajena y la tabla que referencia.
REFERENCES Tabla **ON DELETE CASCADE** la ultima parte actualiza una con la otra.
CONSTRAINT nombre **CHECK** (Columna BETWEEN 10 AND 20) Define una restricción (valor entre 10 y 20).
CONSTRAINT nombre **UNIQUE** (Columna) Define que no se repita la fila dentro de la columna.
CONSTRAINT nombre El resto de restricciones si las hay
);
- **CREATE TABLE** Tabla Crea objetos de base de datos. Tablas, vistas, sinónimos, etc
AS SELECT; A partir de una selección con la clausula AS y el SELECT correspondiente.
- **DROP TABLE** Tabla; Elimina la tabla indicada.
- **DROP TABLE** Tabla Elimina la Tabla indicada y las restricciones de integridad referencial que
CASCADE CONSTRAINTS; remitan al la clave primaria en otra tabla o tablas.
- **TRUNCATE TABLE** Tabla; Elimina el contenido de la tabla indicada (pero mantiene la estructura).
- **ALTER TABLE** Tabla Modifica tablas.
ADD (Columna1 valor, Columna2 valor, ..), Añade campos nuevos
MODIFY (Columna1 valor, Columna 2 valor,..), Modifica columnas
DROP COLUMN (Columna1, Columna 2), Elimina columnas
ADD CONSTRAINT nombre **PRIMARY KEY** (Columna1,..), Añade restricciones
DROP CONSTRAINT nombre **PRIMARY KEY** (Columna1,..), Elimina restricciones
DISABLE CONSTRAINT nombre **PRIMARY KEY** (Columna1,..); Des-habilita restricciones
ENABLE CONSTRAINT nombre **PRIMARY KEY** (Columna1,..); Habilita restricciones

```
CREATE TABLE ALUMNOS07
(
  NUMERO_MATRICULA NUMBER(6) NOT NULL,
  NOMBRE VARCHAR2(15) NOT NULL,
  EDAD NUMBER(2) NOT NULL ,
  FECHA DATE DEFAULT SYSDATE,
  DIRECCION VARCHAR2(30),
  LOCALIDAD VARCHAR2(15),
  COD_PROVINCIA NUMBRE(2), NOT NULL,
  CONSTRAINT PK_NUMERO PRIMARY KEY(NUMERO),
  CONSTRAINT EDAD CHECK (EDAD BETWEEN 18 AND 35),
  CONSTRAINT FK_COD_PROVINCIA FOREIGN KEY(COD_PROVINCIA)
  REFERENCES PROVIN ON DELETE CASCADE
);
```


CREATE TABLE COPIAOFICINAS
AS SELECT * FROM OFICINAS ;

DROP TABLE ALUMNOS07;

ALTER TABLE TIENDAS2
ADD CONSTRAINT PK_ELNIF PRIMARY KEY (NIF);

ALTER TABLE USUARIOS
DROP COLUMN CUOTA_FAMILIAR ;

Comando para que liste las tablas del usuario:
SELECT TABLE_NAME FROM USER_TABLES;

Creación VISTAS:

➤ **CREATE VIEW** 'Nombre vista'
AS SELECT Columna1, Columna 2, ...
FROM Tabla
WHERE Condición

➤ **CREATE OR REPLACE VIEW**
'Nombre vista' (Name1, Name2,...)
AS SELECT Columna1, Columna 2, ...
FROM Tabla
WHERE Condición

➤ **CREATE VIEW** 'Nombre vista'
AS SELECT Columna1, Columna 2, ...
FROM Tabla
WHERE Condición

WITH CHECK OPTION

WITH READ ONLY

➤ **DROP VIEW** 'Nombre vista' ;

Crea una vista con el nombre indicado como si de una tabla se tratase

Se definen las columnas que estarán en la vista

Define de que tabla se hace la vista.

Define una condición, para seleccionar determinadas filas.

Crea o Reemplaza una vista creada

asignamos nuevos nombres a las columnas que se seleccionan.

Se definen las columnas que estarán en la vista

Define de que tabla se hace la vista.

Define una condición, para seleccionar determinadas filas.

Crea una vista con el nombre indicado como si de una tabla se tratase

Se definen las columnas que estarán en la vista

Define de que tabla se hace la vista.

Define una condición, para seleccionar determinadas filas.

Esta opción que se pone al final, Sirve para que todas las operaciones que se hagan en la vista cumplan las condiciones con la que se creo, siempre que haya una condición se pondrá.

Esta opción que se pone al final, Sirve para que solo sea de lectura la vista (SELECT) y no se pueda modificar.

Con esta sentencia borramos una vista creada.

CREATE VIEW ACTIVIDADCUOTA3500 AS SELECT CODIGO_ACTIVIDAD, DESCRIPCION FROM ACTIVIDADES
WHERE CUOTA > 3500 ;

Comando para visualizar las vistas del usuario:
SELECT VIEW_NAME, TEXT FROM USER_VIEWS;