UT 4: Realización de Consultas

UT4
REALIZACION DE CONSULTA

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

UT4:REALIZACION DE CONSULTAS

OBJETIVOS

- Conocer el estándar SQL como lenguaje para manipular las tablas y sus datos.
- Utilizar las sentencias, operadores y funciones que constituyen la base de SQL.
- Trabajar con múltiples tablas y vistas.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Tipos de sentencias SQL

TIPOS DE LENGUAJES

- **DML** (Data Manupulation Language)

  Permite manipular y consultar la información que contienen las tablas. (SELECT, INSERT, DELETE, UPDATE,...)

- **DDL** (Data Description Language)

  Crea y mantiene la estructura de la BD. Usada por programadores y administradores. (CREATE, DROP, ALTER,...)

- **DCL** (Data Control Language)

  Permite gestionar el acceso –confidencialiad- (GRANT, REVOKE,..) y las transacciones a la BD (COMMIT, ROLLBACK,...)

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Elementos del Lenguaje SQL

NOMBRES DE COLUMNAS

Para referenciar una columna se puede indicar de las siguientes formas:

- **nombreColumna**
- **nombreTabla.nombreColumna**
- **esquemaUsuario.nombreTabla.nombreColumna**

NOTA: Siempre que la sentencia afecte a más de una tabla es aconsejable especificar la tabla a la que pertenece cada columna

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

SENTENCIA SELECT

SELECT [ALL/DISTINCT]
[expre_colum1, expre_colum2,, expre_colum || *]
FROM [nombre_tabla1,…., nombre_tablaN]
[**WHERE** condición]
[**ORDER BY** expre_colum [**DESC || ASC**] [, expre_colum
[**DESC || ASC] ..];**

expr_colum ➔ columna, una constante, una expresión aritmética una función o varias funciones anidadas

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

SENTENCIA SELECT

- Permite realizar una consulta a la BD

- Según el nivel de complejidad indicado, se puede obtener:
  - una unidad de datos
  - Todos los datos
  - Un subconjunto de datos
  - Cualquier conjunto de subconjunto de datos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

FROM [name_T1, name_T2....., name_Tn]

- Especifica la tabla o lista de tablas de la que se recuperan los datos

- Si el usuario que hace consulta no es el propietario de la tabla es necesario especificar el nombre de usuario delante de ella

  Ejemplo:

  ```
  Select * from Talumnos;
  Select * from PROFESOR.Talumnos;
  ```

- Es posible utilizar **ALIAS** como nombre de una tabla para ello:

  ```
  Select A.nombre,A.nota from PROFESOR.Talumnos A
  ```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

WHERE [condición]

- Permite establecer una condición que se aplicará a la consulta
- Formato de la condición:

  expresión OPERADOR expresión

Operadores de comparación	**Comparación**	**=,>,<,>=,<=,!=,<>**
	lógicos	**AND, OR, NOT**
	Conjunto de valores	**IN, NOT IN, BETWEEN, NOT BETWEEN**
	Cadena de caracteres	**LIKE, _ , %**

- Se permiten condiciones múltiples: AND, OR y NOT

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

WHERE [condición]

Ejemplos:

Select ..... where nota =5;

Select ..... where (nota=5) and (curso='DAI1');

Select ..... where (nota is NULL) or (UPPER (nom_alum) = 'PEDRO');

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ORDER BY [expr_col1 [DESC || ASC]] [,expr_col1 [DESC || ASC]]....

- Especifica el criterio de clasificación del resultado de la consulta
- ASC || DESC especifica el criterio de ordenación. **ASC por defecto**
- Se pueden anidar criterios, siendo el de más a la izquierda el principal

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ORDER BY [expr_col1 [DESC || ASC]] [,expr_col1 [DESC || ASC]]....

Ejemplos:

Select * from ALUMNOS order by notas/5;

Select * from ALUMNOS order by curso DESC, apellidos, nombre;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ALL

- Recupera todas las filas aunque algunas estén repetidas.
- Opción por Defecto

DISTINCT

- Recupera solo las filas que son distintas

Ejemplo:

```
select DISTINCT num_dpto from EMPLEADOS;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

SELECCIÓN DE COLUMNAS

SELECT [ALL/DISTINCT]
[*expre_colum1, expre_colum2,…, expre_colum* || *]
FROM [nombre_tabla1,…., nombre_tablaN]

- Permite seleccionar las columnas a visualizar

Ejemplo:

```
select numemp,nombre,direc,fcha_alta from EMPLEADOS;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ALIAS DE COLUMNAS

SELECT [ALL/DISTINCT]
[*expr_colum1 [as] "**ALIAS_COL1**", expr_colum2 [as] "**ALIAS_COL2**",... || *]*
FROM [nombre_tabla1,…., nombre_tablaN]

- Permite definir alias a las columnas y estos valores serán utilizados como cabeceras de presentación.

Ejemplo:

select numemp "**Num Emp**",nombre"**Nom Emp**",direc "**Direccion**", fcha_alta from EMPLEADOS;

NOTA: *no se pueden utilizar alias en las cláusulas where, order by, group by; será necesario reescribir la expresión*

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

EJERCICIOS

Crear un usuario U4 y lanzar el script U4.sql.

Definir las restricciones existentes en las tablas, teniendo en cuenta que:

- Un empelado solo pertenece a un departamento
- Un departamento puede tener 0 o más empleados
- Un empleado tiene siempre un director, que es otro empleado, excepto el presidente que no tiene ninguno

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

EJERCICIOS

- Seleccionar de la tabla EMPLE todos los empleados del departamento número 20.
- Modificar la consulta anterior para que aparezca ordenada por apellidos
- Modificar la consulta anterior para que se visualice solo el número de empleado, el nombre, su puesto y el número de departamento al que pertenece
- Modificar la consulta anterior para que se muestre como nombre de las columnas en la cabecera:

  Num_Emp Nombre Puesto Num_Dpto

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

EJERCICIOS

- Consulta de los empleados cuyo oficio sea analistas
- Modificar la consulta anterior para que se muestre ordenado por número de empleado
- Seleccionar de la tabla EMPLE aquellas filas del departamento 20 cuyo oficio sea vendedor
- Realizar una consulta de todos los empleados ordenado descendentemente por el puesto que ocupan y dentro de un mismo puesto deberán de aparecer ordenados descendentemente por el nombre
- Consultar cuales son los distintos tipos de Puestos que tienen los empleados

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ENTRADA Y SALIDA DE DATOS POR TECLADO

Para **leer valores por pantalla** se puede usar el comando ACCEPT y las variables de sustitución:

- **Variable de sustitución:** pueden aparecer directamente en la sentencia SELECT sin necesidad de definirla, anteponiéndola el símbolo **&** y SQL nos preguntará el valor que queremos asignarle.

- **ACCEPT** permite declarar una variable de entorno y leer su valor poniendo el mensaje deseado en el Prompt.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ENTRADA Y SALIDA DE DATOS POR TECLADO

ACCEPT variable [NUMBER|CHAR|DATE] [FORMAT]
[PROMPT text] [HIDE]

HIDE → para ocultar lo que se introduce (contraseñas)

Para utilizar la variable accedemos a ella anteponiéndole el símbolo **&**. ***Si la variable se define como ALFANUMÉRICA debe ir encerrada SIEMPRE entre comillas simples***.

PERO: No podemos utilizar ACCEPT para leer variables dentro de un bloque PL/SQL, si queremos utilizarlo debemos hacerlo fuera del mismo.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ENTRADA Y SALIDA DE DATOS POR TECLADO

```
ACCEPT  v_dep PROMPT 'Introduce NUMERO departamento:';
SELECT   *   FROM departamentos WHERE dept_no= &v_dep;

--------------------------------------------------------------------------------

ACCEPT  v_ape PROMPT 'Introduce apellido a buscar:';

//como v_ape es alfanumérico debe de ir entre ' '
SELECT   *   FROM departamentos WHERE apellido= '&v_ape';
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ENTRADA Y SALIDA DE DATOS POR TECLADO

ACCEPT dep PROMPT 'Introduce NUM departamento:';

SELECT * FROM depart WHERE dept_no= &dep;

Ejecución:

Introduce NUM departamento:

10

Select * from depart where dept_no = 10

ACCEPT ape PROMPT 'Introduce apellido a buscar:';

//como ape es alfanumérico debe de ir entre ' '
SELECT * FROM departamentos
WHERE apellido= '&ape';

Ejecución:
Introduce apellido a buscar::

GONZALEZ

SELECT * FROM departamentos
WHERE apellido= **'GONZALEZ'**

Si no ponemos las ' ' , consideraría que GONZALEZ es una columna de la tabla o una variable

SELECT * FROM departamentos
WHERE apellido= **GONZALEZ** → **ERROR**

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

ENTRADA Y SALIDA DE DATOS POR TECLADO

Existe la posibilidad de usar el & como prefijo delante de una variable que no esté definida, en este caso se interpreta la entrada de teclado *pero no se almacena su valor en memoria*.

```
SELECT * FROM EMPLEADO WHERE DPTO = &NUMDPTO;
```

Donde NUMDPTO no esta definida previamente, en este caso nos pedirá el dato en el momento de ejecutarse el comando, *pero este no se almacena en memoria*

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

CONSULTAS DE DATOS

EJERCICIOS

- Consulta de los empleados de un departamento que se introducirá por teclado
- Consulta de los empleados cuyo trabajo sea uno que se introduce por teclado
- Seleccionar todos los trabajadores de un departamento y que realicen un trabajo que se introduce por teclado
- Contabilizar el número de empleados que hay en un departamento y de un oficio que se introducen por teclado, utilizando para ello la expresión COUNT(*) en la parte de la sentencia SELECT que muestra las columnas

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADORES ARITMÉTICOS

- Permiten formar expresiones con constantes,valores de columnas y funciones de valores de columnas.

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División

```
Select NOM_ALUM “Nombre Alumno”, (NOTA1 + NOTA2 + NOTA3)/3  “Nota”
    from NOTAS_ALUMNOS;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADORES DE COMPARACION

Operador	Operación
=	Igual
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
!= <>	distinto

- Formato: **where expr1 operador expr2**

Select NOM_ALUM “Nombre Alumno”, (NOTA1 + NOTA2 + NOTA3)/3 “Nota”
from NOTAS_ALUMNOS where (NOTA1 + NOTA2 + NOTA3)/3 >=5;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADORES LOGICOS

Operador	Operación
AND	True si las dos condiciones son verdaderas
OR	True si una de las condiciones es verdadera
NOT	True si la condición es falsa

- Formato: **where cond1 operador_logico cond2**

```
Select NOM_ALUM “Nombre Alumno”,
    (NOTA1 + NOTA2 + NOTA3)/3 “Nota”
    from NOTAS_ALUMNOS
    where ((NOTA1 + NOTA2 + NOTA3)/3 >=5) and (NOTA1 >=4);
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADOR LIKE y PATRONES DE BÚSQUEDA

- Permite realizar consulta de datos que se ajusten al formato especificado por los caracteres comodín.
- El carácter comodín puede aparecer en cualquier parte del formato.

comodín	Significado
‘%’	Cualquier cadena de 0 o más caracteres
‘_’	Marcador de posición: representa un carácter cualquiera

Formato:

where col1 like ‘formato_cadena’ AND/OR col2 like ‘formato_cadena2’....

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

EXPRESIONES REGULARES

LEER ARTICULO SOBRE EXPRESIONES REGULARES E INCORPORAR DICHA INFORMACIÓN A LAS DIAPOSITIVAS

http://www.oracle.com/technetwork/es/articles/sql/expresiones-regulares-base-de-datos-1569340-esa.html

[n|p]ata

ca[^snt]a

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADOR LIKE y PATRONES DE BÚSQUEDA

Ejemplo

… editorial like ‘P%’ : busca cadenas que comiencen por P;

… editorial not like 'P%’ : busca cadenas que no comiencen P

... like ‘_A%’ : busca cadenas que comiencen con una A en 2º posición

... like ‘%Z’: busca cadenas que acaben en Z

... like ‘___’: busca cadenas que tengan 3 caracteres

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADOR LIKE y PATRONES DE BÚSQUEDA

Para buscar cadenas **que contengan los caracteres usados como comodines en los patrones**, es decir como literales, se debe utilizar la opción ESCAPE indicando el carácter de escape

…..LIKE 'patron \ _ patron' ESCAPE '\';

…..LIKE 'patron \% patron' ESCAPE '\';

... like ‘A\%%’ ESCAPE '\' : busca cadenas comiencen con A% ….

... like ‘%_Z' ESCAPE '\' : busca cadenas que acaben en _Z

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

OPERADOR LIKE

Ejemplo	Significado
LIKE ‘Director’	
LIKE ‘M%’	
LIKE ‘%x%’	
LIKE ‘__M’	
LIKE ‘N_’	
LIKE ‘_R%’	

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

NULL y NOT NULL

- Una columna de una fila es NULL si esta vacía en caso contrario es NOT NULL

Formato:

where columna IS NULL / IS NOT NULL

Ejemplo:

```
select * from ALUMNOS where NOTA1 is null  // alumnos que no tengan nota1

 select * from ALUMNOS
     where NOTA1 is not null //alumnos que tengan un valor numérico en nota1
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

EJERCICIOS

- Obtener un listado ordenado descendentemente por el salario de los empleados que ganan mas de 1500
- Obtener un listado de los empleados que cobran comisión
- Obtener el mismo listado pero utilizando en la comparación NULL
- Obtener un listado de los empleados cuyo nombre empieza por J
- Listado de los empleados cuyo nombre empieza por J o S
- Listado de los empleados que tengan una A en la segunda posición del nombre
- Lista de los todos empleados y comprobar que sucede:

select emp_no,apellido, oficio, salario, comision, (salario + comision) "Nomina" from EMPLE

Num_emp Nombre Puesto Salario Comisión Nomina

NVL(columna,valor): sustituye los valores nulos de la columna por el valor indicado

- Modificar el listado anterior para que aparezca ordenado por Nómina

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

COMPROBACIÓN CON CONJUNTO DE VALORES

- Permite comparar una columna o una expresión con una lista de valores

Operador	Comprueba que:
IN	Una expresión o valor pertenece a un conjunto de valores
BETWEEN	Una expresión o valor esta comprendido entre un rango de valores

Formato:

expresion [NOT] **IN** (valor1,valor2,...)
expresion [NOT] **BETWEEN** valor_inicial AND valor_final

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

OPERADORES

EJERCICIOS. Usando las tablas del esquema del usuario u4

- Consultar los empleados cuyo número de departamento sea 10 o 30
- Consultar los empleados cuyo número de departamento no sea ni 10 ni 30
- Consultar los empleados que ocupen los puestos de ANALISTA, EMPLEADO y VENDEDOR
- Consultar los empleados que no sean ni ANALISTA, ni VENDEDOR ni EMPLEADO
- Obtener un listado de los empleados cuyo salario esta comprendido entre 1000 y 2000 euros
- Obtener aquellos que no se encuentren entre ese rango
- Obtener el apellido, salario y dept_no de aquellos cuyo salario sea mayor de 2000 en los departamentos 10 o 20.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

Consiste en utilizar el valor resultante de una consulta como elemento de otra consulta

```
SELECT ....
    FROM ....
    WHERE  columna  operador  (SELECT....
                                    FROM...
                                    WHERE ......);
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

SUBCONSULTAS QUE GENERAN <u>VALORES SIMPLES</u> (se puede usar = o IN)

- Devuelven una sola fila o un valor simple
- En caso de que devuelva más de un valor se producirá un mensaje de error

```
Select   talum_nombre
   from ALUMNOS
            where talum_curso =
                     (select tasig_curso from ASIGNATURA
                              where tasig_cod_asig = 'PLE');
```

2º

1º

Donde nº corresponde al orden en el que se evaluará la sentencia

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

SUBCONSULTAS QUE GENERAN <u>LISTAS DE VALORES</u> (solo se puede usar IN)

- Devuelven más de una fila o más de un valor
- En este caso utilizaremos el operador IN / NOT IN en la cláusula WHERE

2º
```
Select   talum_nombre      from ALUMNOS
                           where talum_asig IN
                                 (select tasig_cod_asig from ASIGNATURA
                                       where tasig_curso = '1DAI');
```
1º

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

EJERCICIOS

- Obtener el nombre, oficio y departamento de los empleados que tiene el mismo puesto que SALA
- Obtener el nombre, oficio y número de departamento de los empleados que se encuentran en MADRID y BARCELONA, sabiendo que estos son valores de la tabla DEPART
- Obtener un listado con el nombre, oficio, salario y numero de departamento de los empleados con el mismo oficio y salario que FERNÁNDEZ
- Obtener un listado con el nombre, oficio y numero de departamento de los empleados con el mismo oficio que MARTÍN y cuyo salario se encuentre entre 1000 y 2000 euros.
- Obtener los nombres de los compañeros que trabajan en el mismo departamento que SALA y GIL

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

SUBCONSULTAS CORRELACIONADAS

- Es aquella que hace referencia a una columna o varias de la consulta más externa.
- hacen un proceso fila a fila, de modo que la subconsulta se ejecuta una vez por cada fila de la consulta principal.

Ej. Empleados cuyo sueldo supera la media de su departamento

```
select * from  EMPLE  E
      where SALARIO > ( select avg(SALARIO) from EMPLE
                                    where DEPT_NO = E.DEPT_NO);
```

avg(columna) → calcula la media de los valores no null de esa columna

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

SUBCONSULTAS

SUBCONSULTAS CORRELACIONADAS

Las subconsultas correlacionadas hacen un proceso fila a fila, de modo que la subconsulta se ejecuta una vez por cada fila de la consulta principal. Esto es absolutamente diferente respecto a la ejecución normal de una subconsulta, ya que normalmente la subconsulta clásica se ejecuta primero, y con sus resultados se ejecuta la consulta principal,

```
select * from emple
where dept_no = (select dept_no from emple where dnombre='VENTAS')
```

En las correlacionadas la subconsulta se realiza cada vez por cada una de las filas de la consulta principal , comparando en el ejemplo si el salario medio de ese departamento es mayor o no, y en caso de que se cumpla, lo muestra y pasaría a la siguiente fila

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

COMBINACION DE TABLAS

Podemos combinar varias tablas para ello se tiene que tener en cuenta las siguientes reglas:

- Es posible unir tantas tablas como deseemos
- la cláusula SELECT pueden citar columnas de todas las tablas
- Si existen columnas con el mismo nombre de distintas tablas, identificarlas con **nomTabla.nomColumna**
- El criterio que se siga para combinar las tablas ha de especificarse en la cláusula WHERE. Si se omite el resultado será el producto cartesiano de las dos tablas

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

COMBINACION DE TABLAS

Cuando combinamos tablas. PRODUCTO CARTESIANO

Tabla: CAMISAS

Tabla: PANTALONES

Producto cartesiano CAMISAS * PANTALONES
sería todas las posibles combinaciones de cada una de las camisas con todos los pantalones, aunque algunas realmente no son aconsejables

Y si incorporamos una tabla CALZADO (con deportivas, botas, zapato tacón, zapato plano)
CAMISAS * PANTALONES * CALZADO

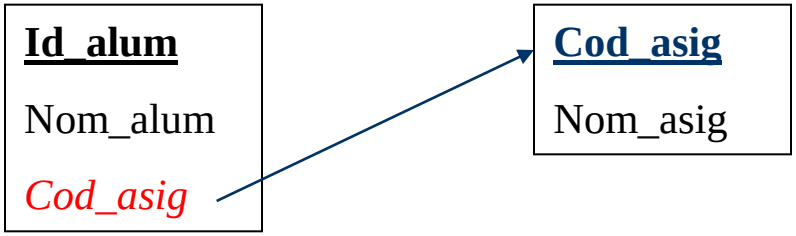
Deberíamos de introducir una condición (WHERE) que permita seleccionar aquellas combinaciones que se ajustan a la estética (que están relacionados), evitar las combinaciones que no tienen sentidos (rayas con cuadros,….etc)

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

COMBINACION DE TABLAS

¿Cuál es esa condición?

Aquella que permita relacionar las tablas y dar sentido a la relación



Select alumnos.*, asignaturas.* from alumnos, asignaturas
where **alumnos.cod_asig** = **asignaturas.cod_asig**;

Selecciona de todas las filas del producto cartesiano aquellas que cumplan esta condición, que en nuestro caso, son los que están relacionados, y son aquellas que se unen por la clave ajena.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

COMBINACION DE TABLAS

EJERCICIOS

- Mostrar un listado con la siguiente información

Num Emp	Nombre	Puesto	Num_Dpto	Nom_Dpto
--------	--------	--------	------	----------

- Modificar el listado anterior para que aparezca también el nombre de la ciudad.

Num Emp	Nombre	Puesto	Nom_Dpto	Ciudad
--------	--------	--------	----------	-------

- Modificar el listado anterior para que se muestre el salario, la comisión y la nómina que cobrará el empleado.

- Obtener el mismo listado solo para los empleados que trabajen en SEVILLA, MADRID, BILBAO

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones

OBJETIVOS

- Identificar las distintas funciones que se pueden usar con la claúsula SELECT
- Utilizar las diferentes funciones para hacer consultas a la base de datos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Introducción

- **FUNCIONES:**
  - ¿Donde se utilizan?: en expresiones
  - ¿Sobre qué actúan? : campos, variables o constantes
  - Resultados que generan:
    - Modificación de la información original

      UPPER(nom)
    - Indica alguna característica de la información
  - Se permiten el anidamiento de funciones

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Introducción

TIPOS DE FUNCIONES

- Funciones Aritméticas
- Funciones de caracteres
- Funciones de manejo de fechas
- Funciones de conversión
- Otras funciones

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

- Trabajan con datos de tipo NUMBER

- En función al grupo de valores numéricos con los que trabajan tenemos 3 tipos de funciones:
  - Funciones de valores simples
  - Funciones de grupos de valores
  - Funciones de lista de valores

NOTA: En ORACLE existe una tabla llamada DUAL, para probar funciones y hacer cálculos rápidos.

DESC DUAL;
Select funcion(valor) from DUAL;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES

- Son funciones sencillas que trabajan con valores simples: un número, variable, una columna de una tabla

FUNCION	PROPÓSITO
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, expresion)	Sustituye el valor nulo por el valor indicado en expresión

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES

FUNCION	PROPÓSITO
POWER(m, exponente)	Calcula la potencia de un número
ROUND(n [,m])	Redondea “n” con el número de dígitos de precisión “m”. Si m<0 redondea los “m” dígitos a la izquierda del punto decimal
SIGN(n)	Signo del valor “n” (SI N<0 ➔ -1, SI N>0➔ 1, 0 si N=0)
SQRT(n)	Devuelve la raíz cuadrada de “n”
TRUNC(n [,m])	Trunca “n” para que tenga “m” dígitos decimales de precisión. Si m<0 trunca los “m” dígitos a la izquierda del punto decimal
VARIANCE(valores)	Devuelve la varianza de un conjunto de valores

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES: ejemplos

FUNCION	valor
ABS (-20)	20
CEIL(30.9)	31
CEIL(-30.7)	-30
FLOOR(30.9)	30
FLOOR(-30.7)	-31
MOD(20,3)	2
NVL(comm,0)	0
POWER(2,3)	8

FUNCION	valor
POWER(4.5,2.4)	36.95
ROUND(1.678,1)	1.7
ROUND(1.3244)	1
ROUND(132.5,-2)	100
ROUND(167.45,-1)	170
SIGN(-10)	-1
SIGN(10)	1
TRUNC(1.546,2)	1.54

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES: Ejercicios

Calcular el valor final del número una vez aplicada la función:

FUNCION	valor
ABS (-32)	
CEIL(20.7)	
CEIL(20.2)	
CEIL(16)	
CEIL(-20.7)	
CEIL(-20.2)	
CEIL(-16)	
FLOOR(20.7)	

FUNCION	valor
FLOOR(20.2)	
FLOOR(16)	
FLOOR(-20.7)	
FLOOR(-20.2)	
FLOOR(-16)	
MOD(11,4)	
MOD(10,15)	
POWER(4,2)	

FUNCION	valor
ROUND(1.5634,1)	
ROUND(1.5634)	
ROUND(1.2234,2)	
ROUND(1.2676,3)	
ROUND(145.5,-1)	
ROUND(145.5,-2)	
ROUND(145.5,-3)	
ROUND(141,-1)	

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES: Ejercicios

Calcular el valor final del número una vez aplicada la función:

FUNCION	valor
ABS (-32)	32
CEIL(20.7)	21
CEIL(20.2)	21
CEIL(16)	16
CEIL(-20.7)	-20
CEIL(-20.2)	-20
CEIL(-16)	-16
FLOOR(20.7)	20

FUNCION	valor
FLOOR(20.2)	20
FLOOR(16)	16
FLOOR(-20.7)	-21
FLOOR(-20.2)	-21
FLOOR(-16)	-16
MOD(11,4)	3
MOD(10,15)	10
POWER(4,2)	16

FUNCION	valor
ROUND(1.5634,1)	1.6
ROUND(1.5634)	2
ROUND(1.2234,2)	1.22
ROUND(1.2676,3)	1.268
ROUND(145.5,-1)	150
ROUND(145.5,-2)	100
ROUND(145.5,-3)	0
ROUND(141,-1)	140

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES: Ejercicios

FUNCION	valor
SIGN(-20)	
SIGN(20)	
TRUNC(1.5634,1)	
TRUNC(1.684,2)	
TRUNC(1.1684)	
TRUNC(187.98,-1)	
TRUNC(187.98,-2)	
TRUNC(187.98,-3)	

Comprobar los resultados utilizando la tabla DUAL

Ej: Select ceil(20,7), ceil(20.2) from dual;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE VALORES SIMPLES: Ejercicios

FUNCION	valor
SIGN(-20)	-1
SIGN(20)	1
TRUNC(1.5634,1)	1.5
TRUNC(1.684,2)	1.68
TRUNC(1.1684)	1
TRUNC(187.98,-1)	180
TRUNC(187.98,-2)	100
TRUNC(187.98,-3)	0

Comprobar los resultados utilizando la tabla DUAL

Ej: Select ceil(20,7), ceil(20.2) from dual;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE GRUPOS DE VALORES

- Son funciones que actúan sobre grupos de valores o columnas de una tabla, por ejemplo funciones estadísticas.

FUNCION	Propósito
AVG(n)	Calcula el valor medio de “n” ignorando los valores nulos
COUNT(* | expr)	Cuenta el número de veces que la expresión evalúa con datos no nulos. El “*” cuenta ***todas*** las filas
MAX(expr)	Calcula el máximo valor de la expresión
MIN (expr)	Calcula el mínimo valor de la expresión
SUM (expr)	Obtiene la suma de valores de la expresión

- La función COUNT admite la cláusula DISTINCT

  COUNT (* | [DISTINCT | ALL] expresion) → contaría filas no duplicadas

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE GRUPOS DE VALORES: Ejemplos

FUNCION	ejemplo
AVG(n)	Select AVG(edad) from nombres;
COUNT(* | expr)	Select COUNT(num_matric) from ALUMNOS;
MAX(expr)	Select MAX (edad) from emp;
MIN (expr)	Select MIN (edad) from emp;
SUM (expr)	Select SUM (unid_vend) from ventas;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE GRUPOS DE VALORES: Ejercicios

- Calcular el salario medio de los empleados
- Calcular el salario medio de los empleados del departamento 10
- Calcular el número de filas de la tabla de empleados
- Calcular el número de filas donde la comisión es nula
- Cual es el salario más alto de la tabla de empleados
- Visualizar el nombre, departamento, puesto y salario del empleado que más cobra.
- Realizar el mismo proceso para conocer el que menos cobra
- Calcular el importe en dinero de la nómina de este mes de todos los empleados
- Calcular el número de oficios que hay en la tabla empleados
- Calcular el número de oficios que hay en los departamentos 10 y 20

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE LISTAS

- Trabajan sobre un grupo de columnas dentro de una misma fila.
- Comparan los valores de cada una de las columnas en el interior de una fila para obtener el mayor o menor valor de la lista.

FUNCION	Propósito
GREATEST (valor1,valor2,...)	Obtiene el mayor valor de la lista
LEAST (valor1,valor2,...)	Obtiene el menor valor de la lista

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones Aritméticas

FUNCIONES DE LISTAS: Ejemplos

```
Select nom_al,GREATEST(nota1,nota2,nota3)  from alumnos;

Select nom_al,LEAST(nota1,nota2,nota3)  from alumnos;

Select greatest ('Benito','Jorge','Isabel') from DUAL;

Select least('Benito','Jorge','Isabel') from DUAL;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES DE CADENA DE CARACTERES

- Trabajan con datos de tipo CHAR o VARCHAR2
- Permiten manipular cadenas de letras u otros caracteres
- Se pueden dividir en dos tipos de funciones:
  - Funciones que devuelven valores carácter
  - Funciones que devuelven valores numéricos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES

- Devuelven un carácter o un conjunto de caracteres.

FUNCION	PROPÓSITO
CHR(n)	Devuelve el carácter cuyo valor binario es el equivalente a “n”
CONCAT(cad1,cad2)	Devuelve cad1 concatenada con cad2
LOWER(cad)	Convierte la cadena a minúsculas
UPPER(cad)	Convierte la cadena a mayúsculas
INITCAP(cad)	Convierte la cadena a tipo título
LPAD(cad1, n [,cad2])	Añade a la izquierda, los caracteres indicados en cad2, hasta que cad1 tenga “n” caracteres de longitud
RPAD(cad1,n [,cad2])	Añade a la derecha, los caracteres indicados en cad2, hasta que cad1 tenga “n” caracteres de longitud

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES

FUNCION	PROPÓSITO
LTRIM(cad [, set])	Suprime un conjunto de caracteres (set) a la izq de la cadena
RTRIM(cad [, set])	Elimina caracteres indicados por set a la dcha de la cadena
REPLACE (cad, cad_busq [, cad_sustitución])	Sustituye un carácter o caracteres de una cadena con 0 o mas caracteres. Si no se pone cad_sustit se supone NULL
SUBSTR(cad,inicio [,n])	Obtiene un subcadena de “n” caracteres desde la posición “inicio” de la cadena. Si “inicio” es negativo la posición de inicio se calcula empezando desde la derecha hacia atrás
TRANSLATE(cad1,cad2,cad3)	Convierte los caracteres que se indican en cad2 por el correspondiente carácter que aparece en la misma posición en cad3 dentro de la cadena cad1. Si hay menos caracteres en cad3 que en cad2 se completa con null

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES: Ej.

FUNCION	valor
CHR(65)	
CONCAT('Hola','mundo')	
LOWER('HOLA')	
UPPER('hola')	
INITCAP('hola mundo')	
LPAD('X',5,'*')	
LPAD('X',10,'>+')	
RPAD('X',5,'*')	
RPAD(''X',10,'>+')	

FUNCION	valor
LTRIM(' hola')	
LTRIM('hola', 'h')	
RTRIM('hola ')	
RTRIM('hola', 'la')	
REPLACE ('blanco negro,'o','a')	
REPLACE ('blanco negro,'o')	
SUBSTR('ABCDEFG',3,2)	
SUBSTR('ABCDEFG',-3,2)	
TRANSLATE('SQLPLUS','SQLU',123)	

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES: Ej.

FUNCION	valor
CHR(65)	'A'
CONCAT('Hola','mundo')	'Hola mundo'
LOWER('HOLA')	'hola'
UPPER('hola')	'HOLA'
INITCAP('hola mundo')	'Hola Mundo'
LPAD('X',5,'*')	'****X'
LPAD('X',10,'>+')	'>+>+>+>+>+>X'
RPAD('X',5,'*')	'X****'
RPAD(''X',10,'>+')	'X>+>+>+>+>+>'

FUNCION	valor
LTRIM(' hola')	'hola'
LTRIM('hola', 'h')	'ola'
RTRIM('hola ')	'hola'
RTRIM('hola', 'la')	'ho'
REPLACE ('blanco negro,'o','a')	'blanca negra'
REPLACE ('blanco negro,'o')	'blanc negr'
SUBSTR('ABCDEFG',3,2)	'CD'
SUBSTR('ABCDEFG',-3,2)	'EF'
TRANSLATE('SQLPLUS','SQLU',123)	'123P31'

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN VALORES NUMERIC.

- Utilizan cadenas de caracteres y devuelven valores numéricos.

FUNCION	PROPÓSITO
ASCII(cad)	Devuelve el valor ASCII de la primera letra de la cadena “cad”
INSTR(cad1,cad2 [,comienzo [,m]])	Busca un conjunto de caracteres (cad2) en la cadena (cad1) y devuelve la posición de la ocurrencia “m”. La búsqueda se realiza desde ‘comienzo’. Si comienzo <0 la posición de comienzo se calcula a partir del final de cad1
LENGTH(cad)	Devuelve el número de caracteres de la cadena

- ..

ASCII: código de caracteres que utiliza 7 bits para representar los caracteres, cada combinación de 7 bits representa un carácter.

pulsando ALT+ numASCII obtenemos el carácter ALT+65 → A

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES: Ej.

FUNCION	valor
ASCII('A')	
ASCII('DAI')	
INSTR('II VUELTA CICLISTA A TALAVERA','TA',3,2)	
INSTR('II VUELTA CICLISTA A TALAVERA','TA',20)	
LENGTH('HOLA MUNDO')	

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Caracteres

FUNCIONES QUE DEVUELVEN CARACTERES: Ej.

FUNCION	valor
ASCII('A')	65
ASCII('DAI')	68 ➔solo la letra D
INSTR('II ***VUELTA CICLISTA A TA***LAVERA','TA',3,2)	17
INSTR('***II VUELTA CICLISTA A TA***LAVERA','TA',20)	22
LENGTH('HOLA MUNDO')	10

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Fechas

FUNCIONES DE MANEJO DE FECHAS

- Permite manipular valores tipo DATE
- Tiene como formato por omisión ‘DD-MM-YY’
- Los literales de fechas deben de encerrarse entre comillas simples

FUNCION	PROPÓSITO
SYSDATE	Devuelve la fecha del sistema
ADD_MONTHS(fecha,n)	Devuelve la fecha incrementada/decrementada en “n” meses
LAST_DAY(fecha)	Devuelve la fecha del último día del mes que contiene fecha
MONTHS_BETWEEN(f1,f2)	Devuelve la diferencia en meses entre las fechas
NEXT_DAY(fecha,cad)	Devuelve la fecha del día de la semana indicado en cad. Donde cad=‘LUNES’,’MARTES’,....
EXTRACT(valor FROM fecha)	Extrae un valor de una fecha concreta. El valor puede ser day, month, year, hours, etc

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Fechas

FUNCIONES DE MANEJO DE FECHAS : Ejemplo

FUNCION	valor
SYSDATE	
ADD_MONTHS(sysdate,2)	
ADD_MONTHS(sysdate,-2)	
LAST_DAY(sysdate)	
MONTHS_BETWEEN(sysdate,'04-06-2006')	
Calcular la edad que tienes	
NEXT_DAY(sysdate,'LUNES')	
EXTRACT(MONTH FROM SYSDATE)	

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN

- Permiten transformar un tipo de dato a otro tipo

FUNCION	PROPÓSITO
TO_CHAR	Transforma un dato tipo DATE o NUMBER a una cadena de caracteres
TO_DATE	Transforma un tipo NUMBER o CHAR en DATE
TO_NUMBER	Transforma un dato tipo cadena en NUMBER

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char. Date➔Char

- TO_CHAR(fecha,'formato'): convierte DATE ➔ VARCHAR2

MASCARA	Formato
cc o scc	Valor del siglo
y,yyy o sy,yyy	Año con coma, con o sin signo
yyyy	Año sin signo
yyy	Últimos 3 dígitos del año
yy	Últimos 2 dígitos del año
y	Último dígito del año
q	Núm del trimestre
ww	Núm de la semana del año
w	Núm de la semana del mes

MASCARA	Formato
mm	Número del mes
ddd	Número del día del año
dd	Número del día del mes
d	Número del día de la semana
hh o hh12	Hora (1-12)
hh24	Hora (1-24)
mi	Minutos
ss	Segundos
ssss	Segundos desde la medianoche

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char . Date➔Char

- TO_CHAR(fecha,'formato'): convierte DATE ➔ VARCHAR2

MASCARA	Formato de caracteres
Syear o year	Año ingles en texto
Month	Nombre del mes
Mon	Nombre del mes abreviado. 3 caracteres
Day	Nombre del día
Dy	Nombre del día abreviado. 3 caracteres
a.m. O p.m.	Muestra am o pm
b.c. O a.d.	Indicador antes de cristo o despues de cristo

Ej. SELECT to_char(SYSDATE,'Day') FROM DUAL → nombre del día de la semana

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Diapositiva con Información adicional

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char . Date➔Char

- El formato de fecha viene definido por el parámetro NLS_TERRITORY, que especifica el idioma para el formato de fecha

- Podemos cambiar el valor por omisión para la fecha con el parámetro NLS_DATE_FORMAT, usando la orden ALTER SESSION

ALTER SESSION SET NLS_DATE_FORMAT = 'DD/month/yyyy' **;**

Ejemplos

- **Select to_char(sysdate,'dd, month yyyy') from dual;**
- **Select to_char(sysdate,'"hoy es" day, dd "de" month "de" yyyy') from dual;**

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char. Num➔Char

- **TO_CHAR(número,'formato')**: convierte NUMBER ➔ VARCHAR2

MASCARA	Ejemplo	Formato
9	999	Devuelve el valor con el número especificado de dígitos
0	9990 0999	Muestra un 0 si el valor contiene 0 en dicha posición
$	$9999	Devuelve el valor con el signo $ a la izquierda
B	B999	Muestra un espacio en blanco si el valor es 0
MI	999MI	Si el número es negativo muestra el digno menos después del número
S	S999 999S	Visualiza el signo en la correspondiente posición

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char. Num➔Char

- **TO_CHAR(número,'formato'):** convierte NUMBER ➔ VARCHAR2

MASCARA	Ejemplo	Formato
PR	999PR	Los número negativos se muestran entre <>
L	9999L	Visualiza el símbolo de moneda en la posición indicada
,	99,999	Devuelve la coma en la posición especificada
.	99.99	Devuelve el punto decimal en la posición especificada
V	999V99	Devuelve el valor multiplicado por 10^n , donde n es el número de 9 después la v
RN	RN	Devuelve el valor en número romanos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_char. Num➔Char

- **TO_CHAR(número,'formato'):** convierte NUMBER ➔ VARCHAR2

Ejemplos:

```
Select TO_CHAR(1,'999'), to_char(-1,'999') from dual;
Select TO_CHAR(avg(edad),'999D99') from alumnos;
Select TO_CHAR(sum(impor_vtas),'999G999D99L') from ventas;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Diapositiva con Información adicional

Funciones de Conversión

CONFIGURACION DE PARÁMETROS

- Los caracteres devueltos en los formatos se especifican inicializando una serie de parámetros.
- Estos parámetros se pueden modificar con la sentencia

ALTER SESSION SET parametro='valor'

PARAMETRO	Valor	Descripción
NLS_NUMERIC_CHARACTERS	D,G	Define los caracteres Decimal (D) y el separador de los miles (G) Formato: NLS_NUMERIC_CHARACTERS='DG' NLS_NUMERIC_CHARACTERS=',.'
NLS_ISO_CURRENCY	C	Especifica el símbolo de territorio. 'ESP'
NLS_CURRENCY	L	Especifica la moneda. '€'

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_number

Formato:

TO_NUMBER(cadena,[',formato'])

- Convierte una cadena a tipo NUMBER según el formato especificado.
- La cadena debe de estar formada por números, el carácter decimal o el signo menos a la izquierda.
- No pueden existir espacios ni caracteres.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_number

Ejemplos

Ejemplo	Número
TO_NUMBER('-12345')	-12345
TO_NUMBER('123,99','999D99')	123,99

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Funciones de Conversión

FUNCIONES DE CONVERSIÓN to_date

Formato: **TO_DATE(cadena,'formato')**

- Convierte la cadena a un valor tipo DATE según formato elegido.
- La cadena deberá de ser igual al formato inicializado en NLS_DATE_FORMAT por el sistema

Ejemplo	Número
TO_DATE('01012005')	01/01/2005
TO_DATE('010105')	ERROR NO COINCIDE CON FORMATO POR DEFECTO

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Otras Funciones

DECODE(var,val1,cod1,val2,cod2,.....,val_defecto)

- Comprueba si “var” es igual a cualquier valor de la lista val1,val2,.... y en su caso lo sustituye por su correspondiente cod1,cod2,... en caso contrario por el valor por defecto.

Ejemplo:

```
Select apellido, oficio, DECODE(upper(oficio),'PRESIDENTE',1,'EMPLEADO',2,5)
                                    from empleado;

Select DECODE (TEMA,'Dibujo','Diseño','Labores','Hogar',TEMA) from LIBRERÍA
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Otras Funciones

MD5. Encriptación criptográfica.

- *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5

- es uno de los algoritmos de reducción criptográficos, La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal, aunque a pesar de su amplia difusión actual, la sucesión de problemas de seguridad detectados desde que, en 1996, Hans Dobbertin anunciase una colisión de *hash*, plantea una serie de dudas acerca de su uso futuro.

MD5("Generando un MD5 de un texto") = 5df9f63916ebf8528697b629022993e8

Se utiliza para almacenar contraseñas en el BD

OTRAS ENCRIPTACIONES

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Diapositiva con Información adicional

Otras Funciones

USER

- Devuelve el nombre del usuario actual.

Ejemplo:

```
SHOW USER
```

UID

- Devuelve el identificador de usuario que asignó el Oracle cuando lo creó.

Ejemplo:

```
SHOW UID
select user,uid from dual;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Cláusulas Avanzadas de Selección

OBJETIVOS

- Elegir las cláusulas necesarias para realizar la agrupación de filas
- Usar órdenes precisas para la agrupación de filas
- Utilización de los OUTER-JOIN
- Utilizar correctamente los operadores de conjuntos en una sentencia SELECT
- Entender el concepto de recuperación jerárquica y aplicarlo en una sentencia SELECT

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING

Hasta ahora:

- **Select**...: para la recuperación de filas
- ....**where**: para seleccionar un número de filas
- Hemos utilizado **funciones** sobre estas filas

Ahora podemos pensar:

- aplicar a esa recuperación de filas **AGRUPAMIENTOS**

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING

Agrupa uno o más conjuntos de filas a través de las columnas especificadas y en el orden especificado. Si se indica la cláusula HAVING permitirá establecer una condición de búsqueda para grupos de filas, permitiendo de esta formas especificar que grupos de filas se visualiza.

SELECT..... FROM....
[GROUP BY colum1, colum2,....] [HAVING condicion]
[ORDER BY.....]

Ejemplo:

```
select avg(salario) from emple group by dept_no;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING

Evaluación de las cláusulas del SELECT en tiempo de ejecución

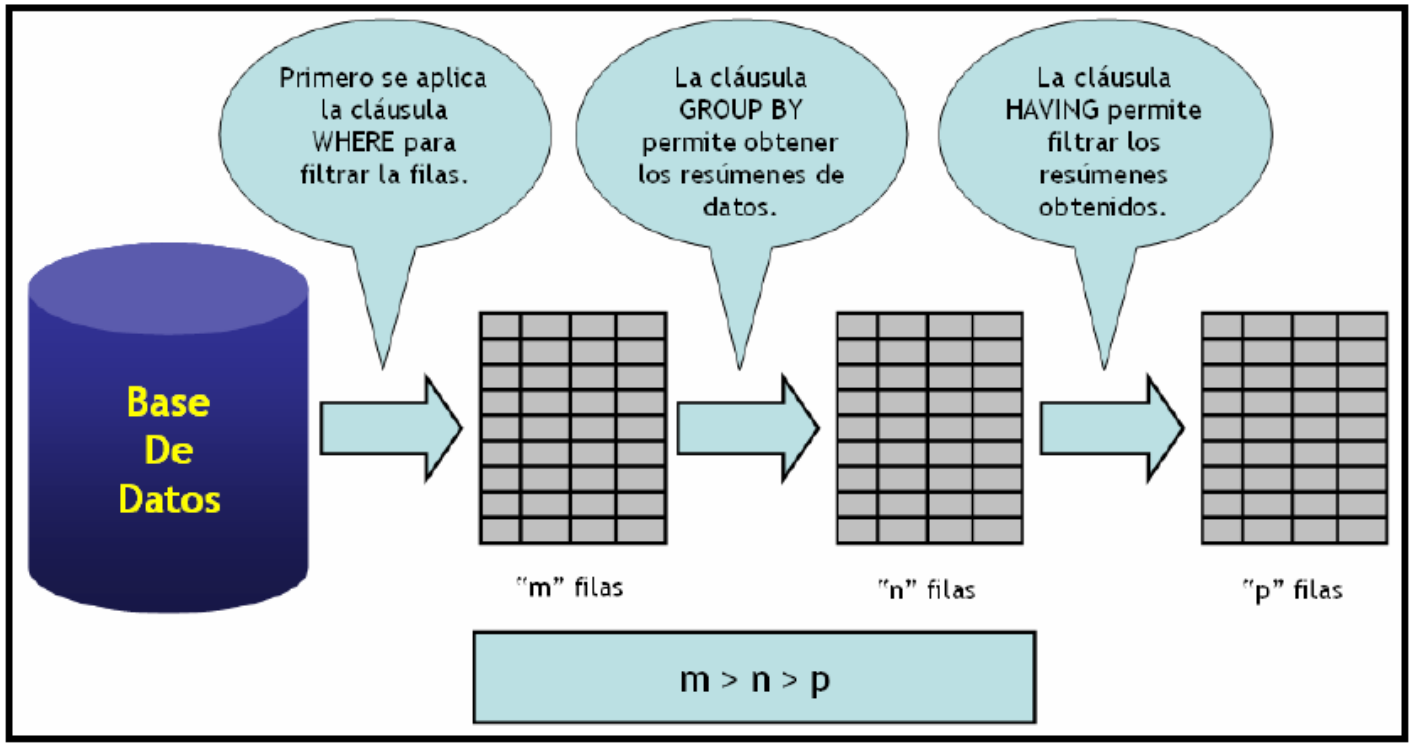
SELECT..... FROM....
WHERE...
[GROUP BY colum1, colum2,....] [HAVING condicion]
[ORDER BY.....]

WHERE	: Selecciona las filas
GROUP BY	: Agrupa estas filas
HAVING	: Filtra los grupos. Selecciona y elimina
ORDER BY	: Clasifica la salida. Ordena los grupos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING.



Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING. Ejemplos

- Visualizar en la tabla EMPLE el número de empleados que hay en cada departamento.
- Modificar la sentencia anterior para que aparezca ordenado por departamento.
- Visualizar en la tabla EMPLE todos los departamentos en los que hay más de 4 empleados.
- Visualizar el listado anterior descendentemente en función al número de empleados que aparecen en cada departamento.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING. Ejemplos

- select dept_no,count(*) from emple group by dept_no;
- select dept_no,count(*) from emple group by dept_no order by DEPT_NO
- select dept_no,count(*) from emple group by dept_no having count(*)>4;
- select dept_no,count(*) from emple group by dept_no having count(*)>4 order by count(*) DESC;

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING. Ejemplos

- Obtener la suma de salarios, el salario máximo y el salario mínimo de cada departamento con salida formateada.
- Obtener los nombres de los departamentos que tengan más de 4 personas.
- Calcular el número de empleados que realizan cada OFICIO en cada DEPARTAMENTO. Los datos a visualizar son: Departamento, Oficio y Número de Empleados.
- Cual es el número de empleados del departamento que más empleados tiene.
- Visualizar el número de departamento, el nombre de departamento y el número de empleados del departamento con más empleados.
- Obtener el número de empleados por departamento que cobran mas de 1500€ en cada departamento

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING. Ejemplos

- Obtener la suma de salarios, el salario máximo y el salario mínimo de cada departamento con salida formateada.

```
select dept_no "DEPARTAMENTO",
  to_char(sum(salario),'9G999G999D99') "TOTAL SALARIOS ",
  to_char(max(salario),'9G999G999D99') "SALARIO MAX",
  to_char(min(salario),'9G999G999D99') "SALARIO MIN"
                                  from emple group by dept_no;
```

- Obtener los nombres de los departamentos que tengan más de 4 personas.

```
select dept_no, dnombre from depart
  where   dept_no in  (select dept_no
                                  from emple
                                  group by dept_no HAVING COUNT(*) >4);
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Agrupación de elemento

GROUP BY y HAVING. Ejemplos

- Calcular el número de empleados que realizan cada OFICIO en cada DEPARTAMENTO. Los datos a visualizar son: Departamento, Oficio y Número de Empleados.

```
select dept_no, oficio, count(*) from emple group by dept_no,oficio;
```

- Cual es el número de empleados del departamento que más empleados tiene.

```
select max(count(*)) from emple group by dept_no;
```

- Visualizar el número de departamento, el nombre de departamento y el número de empleados del departamento con más empleados

```
select d.dept_no, d.dnombre, count(*) from emple e, depart d
where e.dept_no=d.dept_no group by  d.dept_no,d.dnombre
having count(*) = (select max(count(*)) from emple group by dept_no);
```

- Obtener el número de empleados por departamento que cobran mas de 1500€ en cada departamento

```
select dept_no, count(*) from emple where salario>1500
                group by dept_no;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación interna

PRODUCTO CARTESIANO

- El producto cartesiano de dos tablas permite obtiene una tabla con las **columnas** de la **primera tabla unidas** a las **columnas** de la **segunda tabla**, y las filas de la tabla resultante son **todas las posibles concatenaciones** de **filas** de la **primera tabla** con filas de la **segunda tabla**.

```
SELECT * FROM pedidos,clientes
```

Se obtendría todas las posibles combinaciones de pedidos con empleados. Si queremos hacer un selección de esta tabla de aquellos pedidos relacionados con clientes, tendríamos que seleccionar aquellos cuyos departamentos sean iguales (relación clave primaria y ajena)

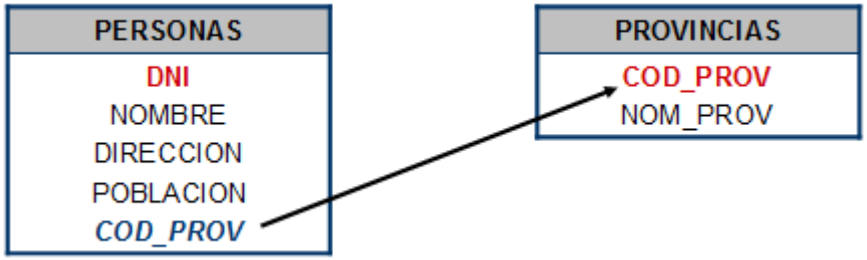
```
SELECT  * FROM pedidos,clientes WHERE pedidos.clie=clientes.numclie
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación interna

PRODUCTO CARTESIANO

Ejemplo




```
SELECT  * FROM personas, provincias
     WHERE personas.cod_prov = provincias.cod.prov
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación interna

INNER JOIN. Notación ANSI

- Si en la relación de tablas *una de las columnas es clave principal y esa columna aparece en la otra tabla*, es más eficiente utilizar otro tipo de composición, el **INNER JOIN** y su resultado sería el mismo que el producto cartesiano sobre dicha columna

select …. from tabla1 NATURAL [INNER] JOIN tabla2

Utilizando esta notación, las columnas comunes solo aparecen una vez

select * from emple natural inner join depart

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación interna

INNER JOIN…USING Notación ANSI

- Si en la relación de tablas *una de las columnas es clave principal y las columnas relacionadas tienen el mismo nombre*, otra formar sería utilizar el USING

```
select …. from tabla1 [INNER]
        JOIN tabla2 USING (campo1)
        JOIN tabla 3 USING (campo 2) ...
        WHERE…
```

Las columnas relacionadas tienen el mismo nombre

```
select * from emple inner join depart  USING (dept_no)
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación interna

INNER JOIN…ON Notación ANSI

- Si en la relación de tablas una de las columnas es clave principal, *pero las columnas relacionadas no tienen el mismo nombre*

```
select …. from tabla1
   [INNER] JOIN tabla2  on (tabla1.col = tabla2.col)
   [INNER] JOIN tabla3  on (tabla2.col = tabla3.col)…
   WHERE…..
```

Las columnas relacionadas no tienen el mismo nombre

```
select * from emple e inner join depart  d on e.dept_no = d.num_dept
```

El resultado sería igual que hacer

```
Select * from emple e, depart d where e.dept_no= d.num_dept
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

OUTER JOINS (+). Left Joins / Rigth Joins

- Permite seleccionar filas de una tabla resultante de una combinación aunque éstas no tengan correspondencia. Dependiendo donde se encuentre el Outer Join se denominará Left Join o Rigth join

Fomato:

```
Select tabla1.colum1, tabla1.colum2, tabla2.column1, tabla2.colum2
   from tabla1, tabla2
   where tabla1.colum1 = tabla2.colum1 (+);
```

Seleccionará todas las filas de la tabla1, aunque no tengan correspondencia con las filas de la tabla 2. El resto de las columnas de la tabla2 se rellena con NULL.

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

OUTER JOINS (+). Left Joins / Rigth Joins

Select * from PARALEER;

COD_LIBRO	NOMBRE_LIBRO
100	CIEN AÑOS DE SOLEDAD
200	LOS MITOS GRIEGOS
300	EL CAMINO

select * from LEIDOS;

COD_LIBRO	FECHA
300	20/02/2005
200	20/04/2005

Select p.cod_libro, nombre_libro, fecha from PARALEER P, LEIDOS L
where P.COD_LIBRO=L.COD_LIBRO;

COD_LIBRO	NOMBRE_LIBRO	FECHA
200	LOS MITOS GRIEGOS	20/02/2005
300	EL CAMINO	20/04/2005

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

OUTER JOINS (+). Left Joins / Rigth Joins

```
Select p.cod_libro, nombre_libro, fecha from PARALEER P, LEIDOS L
        where P.COD_LIBRO = L.COD_LIBRO(+);
```

COD_LIBRO	NOMBRE_LIBRO	FECHA
100	CIEN AÑOS DE SOLEDAD	
200	LOS MITOS GRIEGOS	20/02/2005
300	EL CAMINO	20/04/2005

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

CLAVE PRIMARIA COMPUESTA

Para realizar un outer join (+) donde en una de las tablas la clave es compuesta y de la cual es necesario hacer una consulta previa se realizará de la siguiente forma:

select .. from T1, ***(select .... T2(tabla con clave compuesta).....) T_AUX***
where T1.campo=T_AUX.campo(+) and T1.campo2=T_AUX.campo2(+)

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

OUTER JOINS. Notación ANSI

Sería el equivalente ANSI a la notación + y pueden ser de tres tipos

- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

LEFT OUTER JOINS. Notación ANSI

- Formato:

```
Select … from tabla1 LEFT OUTER JOIN tabla2
                    ON tabla1.campo = tabla2.campo
Select… from tabla1  natural LEFT JOIN tabla2
Select … from tabla1 LEFT JOIN tabla2 using  (campo)
```

Equivalente a

```
Select … from tabla1 t1, tabla2 t2
                    where t1.campo = t2.campo (+)
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

RIGHT OUTER JOINS. Notación ANSI

- Formato:

```
Select… from tabla1  natural RIGHT JOIN tabla2

Select … from tabla1 RIGHT JOIN tabla2 using  (campo)

Select … from tabla1 RIGHT OUTER JOIN tabla2
                    ON tabla1.campo = tabla2.campo
```

Equivalente a

```
Select … from tabla1 t1, tabla2 t2
                    where t1.campo (+) =  t2.campo
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

FULL OUTER JOINS. Notación ANSI

- Formato:

```
Select … from tabla1 FULL OUTER JOIN tabla2
        ON tabla1.campo = tabla2.campo
```

Equivalente a

```
Select … from tabla1 t1, tabla2 t2
                where t1.campo  =  t2.campo (+)
UNION
Select … from tabla1 t1, tabla2 t2
                where t1.campo (+)  =  t2.campo
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

SELF JOINS. Consultas Autoreferenciadas

Hacen un Join consigo mismo

```
select e1.emp_no, e1.apellido, e2.apellido, e1.dir from emple e1, emple e2
where e1.dir = e2.emp_no order by e1.apellido;
```

Emp_no	Apellido	Dir	Jefe
7876	ALONSO	7788	GIL
7499	ARROYO	7698	NEGRO
7782	CEREZO	7839	REY
7902	FERNÁNDEZ	7566	JIMÉNEZ
7788	GIL	7566	JIMÉNEZ
7900	JIMENO	7698	NEGRO
7566	JIMÉNEZ	7839	REY
7654	MARTÍN	7698	NEGRO
7934	MUÑOZ	7782	CEREZO
7698	NEGRO	7839	REY
7521	SALA	7698	NEGRO
7369	SÁNCHEZ	7902	FERNÁNDEZ
7844	TOVAR	7698	NEGRO

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Combinación externa

SELF JOINS. Notación ANSI

Formato:

```
select … tabla1 t1 inner join tabla1 t2
        on t1.campo1 = t2.campo2
        ….
```

Para el ejemplo anterior

```
select e1.emp_no, e1.apellido, e2.apellido “JEFE”, e1.dir
    from emple e1 inner join  emple e2
    on  e1.dir = e2.emp_no order by e1.apellido;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Ejercicios

Ejercicios

Ejecutar el script TABLAS_XXX.SQL, creando un esquema con tu nombre.

1. Para las tablas de profesores y centros, obtener para cada centro el número de empleados. Si el centro carece de empleados, ha de aparecer un 0 como número de empleados

2. Contabilizar el número de empleados que hay en cada departamento y visualizar el número de departamento, nombre y número de empleados, incluidos aquellos que no tengan empleados

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Ejercicios

Ejercicios

1. Para las tablas de profesores y centros de tu usuario, obtener para cada centro el número de empleados. Si el centro carece de empleados, ha de aparecer un 0 como número de empleados

```
SELECT C.COD_CENTRO, NOMBRE, COUNT(DNI) "Empleados"
    FROM PERSONAL P, CENTROS C
    WHERE P.COD_CENTRO(+)=C.COD_CENTRO
    GROUP BY C.COD_CENTRO, NOMBRE;
```

2. Contabilizar el número de empleados que hay en cada departamento y visualizar el número de departamento, nombre y número de empleados, incluidos aquellos que no tengan empleados

```
SELECT count(e.emp_no),d.dnombre,d.dept_no from emple e, depart d
Where e.dept_no(+)=d.dept_no
Group by d.dnombre, d.dept_no;
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

UNION, INTERSECT, MINUS

- Son operadores de conjuntos.
- Permiten obtener filas resultantes de combinar los resultados de varios SELECT para obtener un único resultado.

Formato:

SELECT ..... FROM ..... WHERE.....

Operador_de_conjunto

SELECT ..... FROM .... WHERE

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

UNION

Combina los resultados de dos consultas. Las filas duplicadas se reducen a una fila única

Formato:

SELECT col1, col2,.... FROM tabla1 WHERE condicion

UNION [ALL]

SELECT col1, col2,.... FROM tabla2 WHERE condicion;

Si se utiliza la opción **ALL**, aparecen las filas duplicadas.

Ejemplo: **select nombre from alum UNION select nombre from alum_new;**

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

INTERSECT

Devuelve las filas que son iguales en ambas consultas. Todas las filas duplicadas serán eliminadas antes de la generación del resultado final.

Formato:

SELECT col1, col2,.... FROM tabla1 WHERE condicion

INTERSECT

SELECT col1, col2,.... FROM tabla2 WHERE condicion;

Ejemplo: select nombre from alum INTERSECT select nombre from alum_old;

Sería equivalente a:

Select nombre from alum where nombre IN (select nombre from alum_old);

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

MINUS

Devuelve ***las filas que están en la primera select y no están en la segunda***.
Las filas duplicadas del primer conjunto se reducirán a una fila única antes de que empiece la comparación con el otro conjunto

Formato:

```
SELECT col1, col2,.... FROM tabla1  WHERE condicion
MINUS
SELECT col1, col2,.... FROM tabla2  WHERE condicion;
```

Ejemplo: `select  nombre from alum MINUS select nombre from alum_old;`
Sería equivalente a:

```
Select nombre from alum where nombre NOT IN (select nombre from alum_old);
```

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

REGLAS DE USO DE OPERADORES DE CONJUNTOS

Se deben de tener en cuenta las siguientes reglas:

- Las columnas de las dos consultas se relacionan en orden, de izquierda a derecha
- Los nombres de las columnas de la primera sentencia SELECT no tienen por qué ser lo mismos que los nombres de la segunda
- Las SELECT necesitan tener el mismo número de columnas
- Los tipos de datos deben de coincidir, aunque la longitud no tiene que se la misma

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

REGLAS DE USO DE OPERADORES DE CONJUNTOS

Se deben de tener en cuenta las siguientes reglas:

- Los operadores se pueden encadenar
  select ....INTERSECT .... select .... UNION.....select ....
- Los conjuntos se evalúan de izquierda a derecha

- Se permiten la utilización de paréntesis para modificar la precedencia de las sentencias a evaluar
  select ....UNION .... (select .... INTERSEC.....select ....)

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel

Operadores Relacionales

Ejemplos

Select nombre from alum UNION select nombre from alum_new;

Select nombre from alum INTERSECT select nombre from alum_old;

Select nombre from alum MINUS select nombre from alum_old;

Select nombre,edad from alum MINUS select apellidos, salario from emple;

Select nombre, edad from alum MINUS select salario, apellido from emple;

ERRRRRRROOOORRR!!!!!!!, no coinciden los tipos de datos

Roberto Portabales Enríquez
IES Juan Jose Calvo Miguel