

Diseño de Bases de Datos Multimedia



Sesión 1 de Prácticas: repaso sentencia SELECT

Grado en Ingeniería Multimedia



Universitat d'Alacant
Universidad de Alicante



Departamento de
Lenguajes y Sistemas Informáticos



Tipos de datos más usados

- **Char(n):** Cadena de caracteres de longitud fija de longitud n ($1 \leq n \leq 2000$)
- **Varchar(n) :** Cadena de caracteres de longitud variable con un máximo de n ($1 \leq n \leq 4000$)
(varchar2(n) en ORACLE)
- **INT, INTEGER:** números enteros
- **DECIMAL(p,s):** Números con precisión p y escala s ($1 \leq p \leq 64$) ($p \leq s \leq 30$) ($s \geq 0$)
*(number(p,s) en ORACLE,
number sin especificar precisión es equivalente a double)*
- **DATE:** Datos de tipo fecha. Los valores date deben manejarse encerrados entre comillas simples.



Funciones para datos tipo DATE en ORACLE

- **TO_CHAR(fecha [, formato])**
Convierte la fecha de tipo DATE a un valor VARCHAR2 en el formato especificado en "formato"
- **TO_DATE(cadena [, formato])**
Convierte la cadena de caracteres "cadena" de tipo CHAR a un valor de tipo DATE con el formato especificado en "formato"
- **SYSDATE**
Devuelve la fecha actual del sistema
- **ADD_MONTHS(fecha,n)**
Devuelve la fecha especificada con n meses más
- **MONTHS_BETWEEN(fecha1,fecha2)**
Devuelve los meses transcurridos entre fecha1 y fecha2
- Hay muchas más (consultar manual)

IMPORTANTE:

Se deben usar las funciones TO_CHAR y/o TO_DATE para comparar siempre datos del mismo tipo:

`'25/10/2017' > sysdate` → INCORRECTO, funciona pero ORACLE es quien hace la conversión dependiente de como esté configurado el territorio (posibles errores)

`to_date('25/10/2017','dd/mm/yyyy') >= sysdate` → CORRECTO

o

`'25/10/2017' >= to_char(sysdate, 'dd/mm/yyyy')` → CORRECTO



Formatos para las funciones TO_CHAR y TO_DATE

ELEMENTO

SIGNIFICADO

- - / ' . ; : **'texto'** Marcas de puntuación y texto fijo que se reproduce en el resultado
- **D** Día de la semana (1-7)
- **DAY** Nombre del día de la semana (LUNES, MARTES, ...)
- **DD** Día del mes (1-31)
- **DDD** Día del año (1-366)
- **DY** Nombre reducido del día de la semana (LUN, MAR, MIE...)
- **MM** Mes (1-12)
- **MON** Nombre abreviado del mes
- **MONTH** Nombre completo del mes
- **Q** Trimestre del año (1-4)
- **YYYY** Año con 4 dígitos
- **Y,YYY** Año con punto de millar
- **YY** Año con 2 dígitos



REPASO SENTENCIA SELECT SQL



Sentencia SELECT

SELECT [DISTINCT] listaColumnas

FROM listaTablas

[**WHERE** condición para filas]

[**GROUP BY** listaColumnas por las que se quiere agrupar

[**HAVING** condición para los grupos]]

[**ORDER BY** listaColumnas [**ASC** | **DESC**]]

[] significa que es opcional



Repaso JOIN

Al hablar de “**JOIN**” nos referimos a combinar en una consulta filas de dos o más tablas concatenándolas atendiendo a algún criterio. Dependiendo de la condición que se utilice existen **distintos tipos de join**.



Repaso JOIN

- Si las tablas las relacionamos sin condición, el SGBD devuelve el **producto cartesiano** de esas tablas.
- Cuando existe una condición que relaciona ambas tablas y se devuelven sólo las filas que cumplen la condición se habla de **inner join** (o **simple join**). Esta es la forma más usual de vincular las tablas.
- En ocasiones es útil extender el resultado de estos joins, y entonces trabajaremos con **outer joins**. Al trabajar con outer joins, Oracle devolverá todas las filas que cumplan la condición del join más aquellas filas de la tabla marcada (LEFT, RIGHT) para las que no hemos encontrado filas que hayan hecho que se cumpla la condición del join
Dependiendo de por cuál de las tablas queremos extender el resultado, podemos hablar de:
 - extender el resultado con las filas de la tabla que aparece en primer lugar en el FROM, para ello usaremos **LEFT [OUTER] JOIN en la cláusula FROM**.
 - extender el resultado con las filas de la tabla que aparece en segundo lugar, para esto podemos usar **RIGHT [OUTER] JOIN en la cláusula FROM**
 - extender el resultado con las filas de ambas tablas, habrá que usar **FULL [OUTER] JOIN en la cláusula FROM**.



Repaso JOIN

Veamos unos ejemplos con la tabla VENDEDOR y con la tabla PEDIDO.

- **VENDEDOR** (**numvend** NUMBER(4), **nomvend** VARCHAR2(30), **nombrecomer** VARCHAR2(30), **telefono** CHAR(11), **calle** VARCHAR2(30), **ciudad** VARCHAR2(20), **provincia** VARCHAR2(20))
Clave Primaria: (numvend)
- **PEDIDO** (**numpedido** NUMBER(5), **numvend** NUMBER(4), **fecha** DATE)
Clave Primaria: numpedido
Clave Ajena: numvend -> VENDEDOR



Repaso JOIN

**select numvend, nomvend
from vendedor;**

NUMVEND NOMVEND

200 SEVERINO MARTIN MARTINEZ
1 AGAPITO LAFUENTE DEL CORRAL
2 LUCIANO BLAZQUEZ VAZQUEZ
3 GODOFREDO MARTIN MARTINEZ
4 JUANITO REINA PRINCESA
5 JUANITO REINA PRINCESA
6 MANOLO PIEDRA POMEZ
7 MANUEL PEREZ RODRIGUEZ
8 LUISA PINTO HEREDIA
9 CHEMA PAMUNDI
10 GUSTAVO DE BASICA
11 MARIO DUQUE LIZONDO
12 JOSE ANTONIO MARTINEZ JUAN
13 MANUEL GOMEZ SANTISTEBAN
8001 JUAN RODRIGUEZ JUAN
8002 JUAN MARTINEZ GARCIA
8003 LUIS RODRIGUEZ SALA
100 PEDRO GRACIA MORALES
101 SALVADOR PLA GARCIA
102 SOLEDAD MARTINEZ ORTEGA
55 LUIS GARCIA SATORRE
201 MANUEL ORTUÑO LAFUENTE

22 filas seleccionadas.

select numvend from pedido;

NUMVEND

1
1
2
2
1
5
8002

7 filas seleccionadas.

Repaso JOIN

```
select v.numvend, nomvend, numpedido
from vendedor v, pedido p
where v.numvend= p.numvend;
```

NUMVEND	NOMVEND	NUMPEDIDO
1	AGAPITO LAFUENTE DEL CORRAL	1
1	AGAPITO LAFUENTE DEL CORRAL	2
2	LUCIANO BLAZQUEZ VAZQUEZ	3
2	LUCIANO BLAZQUEZ VAZQUEZ	4
1	AGAPITO LAFUENTE DEL CORRAL	5
5	JUANITO REINA PRINCESA	6
8002	JUAN MARTINEZ GARCIA	7

7 filas seleccionadas.

```
select v.numvend, nomvend, numpedido
from vendedor v left join pedido p on
v.numvend= p.numvend;
```

NUMVEND	NOMVEND	NUMPEDIDO
1	AGAPITO LAFUENTE DEL CORRAL	1
1	AGAPITO LAFUENTE DEL CORRAL	2
1	AGAPITO LAFUENTE DEL CORRAL	5
2	LUCIANO BLAZQUEZ VAZQUEZ	3
2	LUCIANO BLAZQUEZ VAZQUEZ	4
3	GODOFREDO MARTIN MARTINEZ	
4	JUANITO REINA PRINCESA	
5	JUANITO REINA PRINCESA	6
6	MANOLO PIEDRA POMEZ	
7	MANUEL PEREZ RODRIGUEZ	
8	LUISA PINTO HEREDIA	
9	CHEMA PAMUNDI	
10	GUSTAVO DE BASICA	
11	MARIO DUQUE LIZONDO	
12	JOSE ANTONIO MARTINEZ JUAN	
13	MANUEL GOMEZ SANTISTEBAN	
55	LUIS GARCIA SATORRE	
100	PEDRO GRACIA MORALES	
101	SALVADOR PLA GARCIA	
102	SOLEDAD MARTINEZ ORTEGA	
200	SEVERINO MARTIN MARTINEZ	
201	MANUEL ORTUÑO LAFUENTE	
8001	JUAN RODRIGUEZ JUAN	
8002	JUAN MARTINEZ GARCIA	7
8003	LUIS RODRIGUEZ SALA	

25 filas seleccionadas



Repaso JOIN

```
select v.numvend, nomvend, numpedido
from vendedor v, pedido p
where v.numvend= p.numvend;
```

NUMVEND	NOMVEND	NUMPEDIDO
1	AGAPITO LAFUENTE DEL CORRAL	1
1	AGAPITO LAFUENTE DEL CORRAL	2
2	LUCIANO BLAZQUEZ VAZQUEZ	3
2	LUCIANO BLAZQUEZ VAZQUEZ	4
1	AGAPITO LAFUENTE DEL CORRAL	5
5	JUANITO REINA PRINCESA	6
8002	JUAN MARTINEZ GARCIA	7

7 filas seleccionadas.

```
select v.numvend, nomvend, numpedido
from vendedor v right join pedido p on
v.numvend= p.numvend;
```

NUMVEND	NOMVEND	NUMPEDIDO
1	AGAPITO LAFUENTE DEL CORRAL	1
1	AGAPITO LAFUENTE DEL CORRAL	2
2	LUCIANO BLAZQUEZ VAZQUEZ	3
2	LUCIANO BLAZQUEZ VAZQUEZ	4
1	AGAPITO LAFUENTE DEL CORRAL	5
5	JUANITO REINA PRINCESA	6
8002	JUAN MARTINEZ GARCIA	7

7 filas seleccionadas.



Usos prácticos de la sentencia SELECT

- La sentencia SELECT de SQL es la que nos permite obtener información de las tablas de nuestra base de datos.
- Nos servirá para leer información de las tablas de la base de datos desde nuestras aplicaciones, mediante programación y drivers de conexión.
- También la usaremos al realizar consultas directas a la base de datos mediante aplicaciones cliente, como en nuestro caso SQL DEVELOPER.