

SQL El lenguaje de Consulta Estructurado

Introducción, Proyección, Ordenación,
Operaciones de Conjunto, Selección

¿Qué es SQL?

- El lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje de control e interacción con un SGBD.
- La primera versión de SQL fue desarrollado por IBM como lenguaje de descripción y manipulación de datos de la base de datos relacional System R a principios de los setenta.
- Permite
 - la creación y modificación de esquemas externos, conceptuales e internos (Data Definition Language)
 - la consulta y el mantenimiento de los datos. (Data Manipulation Language)

Un poco de historia

- 1970 Codd define el modelo relacional
- 1974 Se inicia la construcción de System R
- 1979 ORACLE es la primera BDR comercial con SQL
- 1982 ANSI forma el Comité de Estándares de SQL
- 1986 ISO (International Standards Organization) y ANSI (American National Standards Institute) desarrollaron una versión estándar llamada SQL86 o SQL1
- 1989 2º estándar ANSI/ISO de SQL (SQL/89)

¿Qué es SQL?

- 1992** 3^{er} estándar ANSI/SQL (SQL/92 o SQL-2)
- 1999** Publicación de SQL-3. Añade expresiones regulares, consultas recursivas (relaciones jerárquicas) y disparadores (triggers).
- 2003** Introduce algo de XML y columnas autonuméricas.
- 2005** SQL-2005. Aumenta la integración de XML en SQL.
- 2008** SQL-2008. Se incluyen los disparadores tipo INSTEAD OF y el comando TRUNCATE.

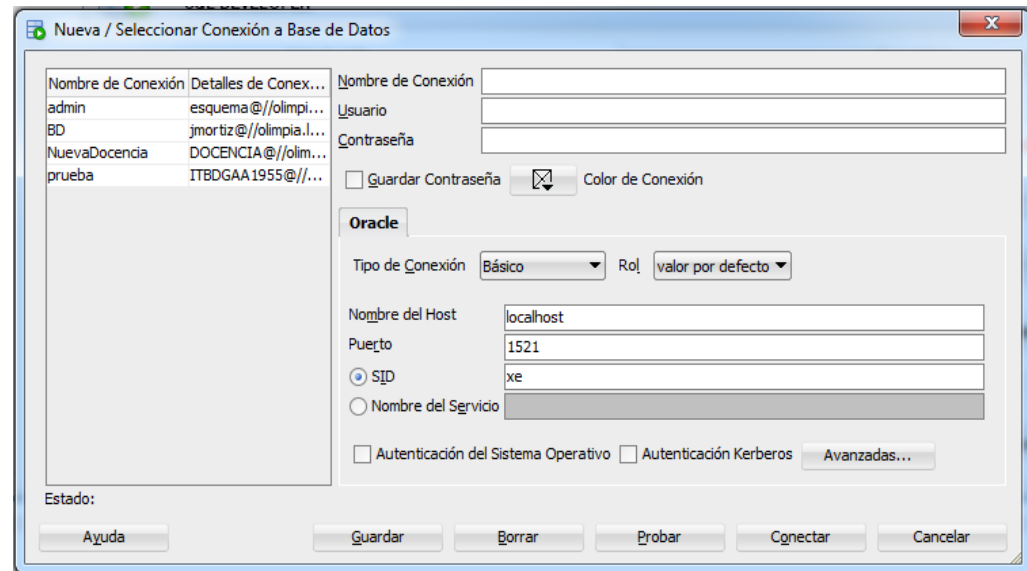
Interactuando con la BD ORACLE



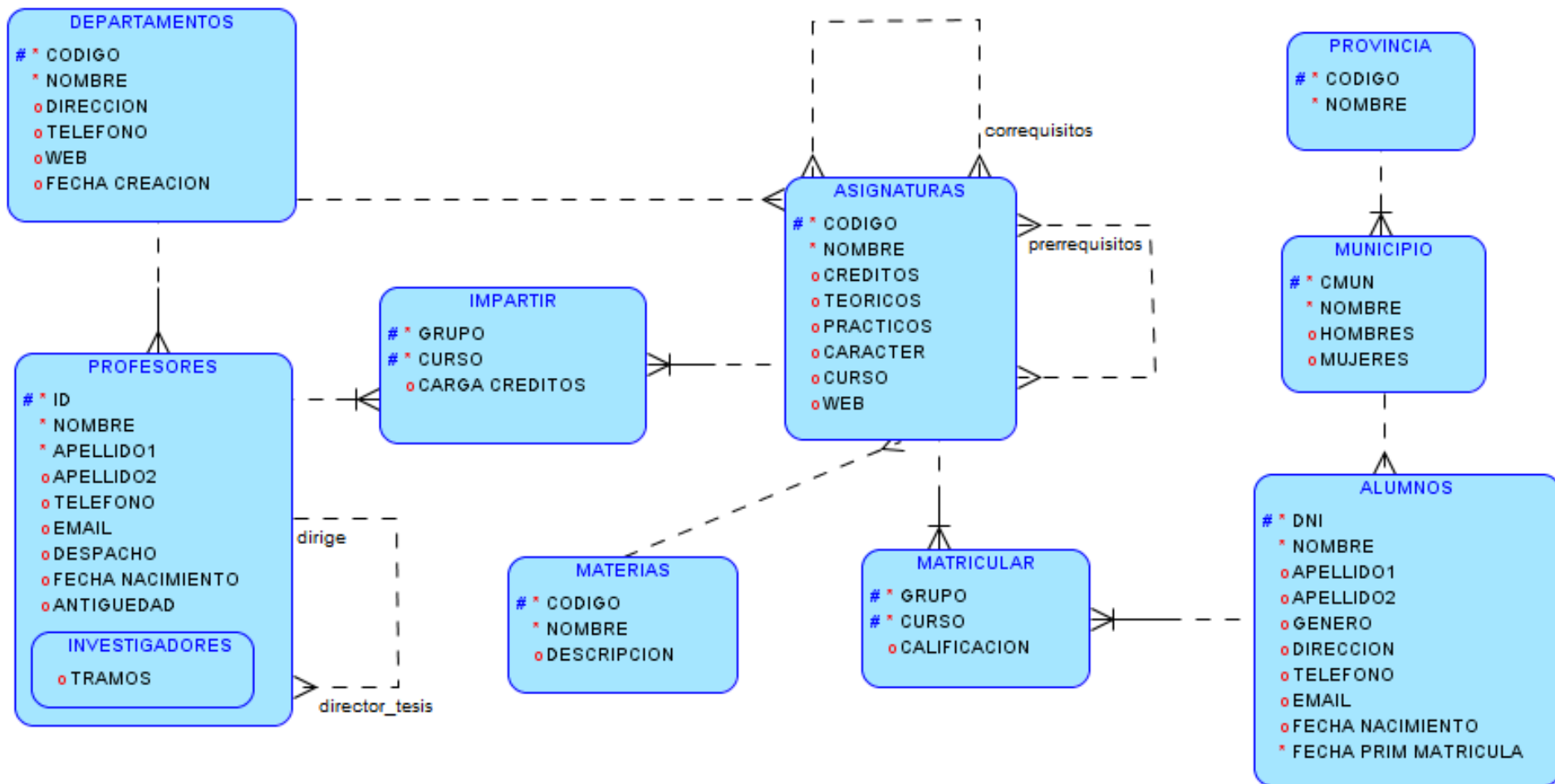
- Arrancar el servidor ORACLE y una instancia de BD
- Uso de un cliente
 - SQL*PLUS (Oracle 8.1.6)
 - SQL*PLUS Worksheet (ORACLE 8i)
 - iSQLPLUS (ORACLE 9i)
 - SQL Developer
 - Otros clientes de terceras partes.

Conexión

- Datos Conexión:
 - Localhost: olimpia.lcc.uma.es
 - SID: apolo
 - Puerto: 1521
- Cuenta general:
 - Usuario: Alumnobd1
 - Clave: bd
- Tablas en usuario: docencia



Base de Datos



Mostrar Información

- La sentencia **SELECT** permite mostrar el cuerpo de una tabla, es decir, el conjunto de filas que están almacenadas en ella.

SELECT * FROM *tabla*

- La sentencia **DESC** permite mostrar el esquema de una tabla, es decir, el conjunto de columnas que la forman y sus dominios.

DESC *tabla*

Mostrar Información

- Mostrar el esquema de la tabla *alumnos*

DESC alumnos;

Nombre	Nulo	Tipo
-----	-----	-----
DNI	NOT NULL	NUMBER(9)
NOMBRE	NOT NULL	VARCHAR2(20)
APELLIDO1		VARCHAR2(20)
APELLIDO2		VARCHAR2(20)
GENERO		VARCHAR2(4)
DIRECCION		VARCHAR2(100)
TELEFONO		VARCHAR2(9)
EMAIL		VARCHAR2(40)
FECHA_NACIMIENTO		DATE
FECHA_PRIM_MATRICULA	NOT NULL	DATE
CPRO		NUMBER(2)
CMUN		NUMBER(5)

Mostrar Información

- Mostrar el cuerpo de la tabla *alumnos*

```
SELECT *  
FROM alumnos;
```

DNI	NOMBRE	APELLIDO1	APELLIDO2	GENERO	DIRECCION	TELEFONO	EMAIL	FECHA_NACIMIENTO
25748955	Guillermo	Villanueva	Pedraza	MASC	C/Rueda 75-1G.	617898999	devott@fairmail.com	22/12/97
87906454	Manuel	Fernandez	Ortega	MASC	C/Latina 15-2G.	617568999	mfo@mimail.com	02/12/98
87675655	Justo	Iglesias	Aramillo	MASC	C/Mirolta 25-1G.	617886999	JIA@fairmail.com	08/12/98
73635423	Rafael	Villanueva	Galvez	MASC	C/Madrid 45-1G.	617884999	Eroto@fairmail.com	20/11/98
73143332	Maria Jose	Lillo	Gamez	FEM	C/Paris 75-1G.	616368999	jmlillo@fairmail.com	18/10/97
23456456	Victoria	Mayordomo	Toro	FEM	C/Quito 75-1G.	628898999	rgueo@micromail.com	12/09/97
67856744	Jose	Gil	Hilo	MASC	C/Tanger 5-1G.	628898999	spider@micromail.com	24/08/97
23423424	David	Alarcon	Garcia y Muntaner	MASC	C/Fuente Paz 25-1G.	619238999	ttguor@micromail.com	09/07/97
67867888	Isidro	Villanueva	Fernandez	MASC	C/Cornisa 35-1G.	617857999	dalarcong@micromail.com	16/07/99
65565643	Nicolas	Bersabe	Alba	MASC	C/Barqueros 45-1G.	952656969	bersabe@micromail.com	30/05/99
97800676	Carlos Alberto	Acosta	Castilla	MASC	C/Zapateros 55-1G.	952398999	acostaCA@mimail.com	30/04/96
45645644	Rodrigo	Castro	Picon	MASC	C/Rodaderos 13-1G.	952893299	rcp56@mimail.com	01/04/96
23423332	Benjamin	Ruiz	Palacio	MASC	C/Italia 32-1G.	951898999	tres@mimail.com	02/03/96
56756777	Gema	Jimenez	Larios	FEM	C/Filipinas 134-1G.	952898119	(null)	03/02/97

...

Proyección

- La proyección sirve para elegir qué atributos queremos en el resultado de una consulta

SELECT Lista_Atributos FROM tabla

- En la lista de atributos se puede especificar un alias para cada atributo.

Atrib1 "Alias completo", Atrib2 "Alias2",...

Proyección

- Listar el Nombre y Código de departamento de todos los profesores

```
SELECT Nombre "NOMBRE DE PILA",  
       Departamento  
FROM profesores;
```

NOMBRE DE PILA	DEPARTAMENTO
Michael	4
Manuel	1
Enrique	2
Miguel	1
Maria del Mar	2
Sergio	1
Carlos	3
Ana	3
Miguel	4
Juana	4
Antonio	4

Eliminación de tuplas repetidas

- Las **tuplas** repetidas en el resultado de la consulta pueden eliminarse con DISTINCT.
- Listar los nombres diferentes de profesores.

```
SELECT DISTINCT Nombre  
FROM profesores;
```

NOMBRE
Michael
Enrique
Sergio
Juana
Miguel
Maria del Mar
Manuel
Antonio
Carlos
Ana

Ordenación

- La cláusula **ORDER BY** permite ordenar las tuplas del resultado de una consulta

ORDER BY Lista_Atributos

- Para cada atributo se puede indicar el sentido de la ordenación:
 - ASC, ascendente (por defecto)
 - DESC, descendente

Ordenación

- Listar el Nombre y Departamento de todos los profesores ordenados por departamento (decreciente) y por orden alfabético dentro del departamento

```
SELECT Nombre, Departamento
FROM profesores
ORDER BY Departamento DESC,
        Nombre ASC;
```

NOMBRE	DEPARTAMENTO
Antonio	4
Juana	4
Michael	4
Miguel	4
Ana	3
Carlos	3
Enrique	2
Maria del Mar	2
Manuel	1
Miguel	1
Sergio	1

Identificación por posición

- Aunque no es recomendable, si la ordenación se realiza utilizando alguna de las columnas de la proyección se puede utilizar su posición en la cláusula ORDER BY.

```
SELECT Nombre, Departamento
FROM profesores
ORDER BY 2 DESC,
        1 ASC;
```

NOMBRE	DEPARTAMENTO
Antonio	4
Juana	4
Michael	4
Miguel	4
Ana	3
Carlos	3
Enrique	2
Maria del Mar	2
Manuel	1
Miguel	1
Sergio	1

Operaciones de conjunto

- Dado que los resultados de las consultas son relaciones (conjuntos de tuplas) se puede utilizar operaciones de conjuntos para combinar los resultados de varias sentencias SELECT.
- Operaciones:
UNION, UNION ALL, INTERSECT, MINUS
- Las relaciones deben ser unión-compatibles.
En caso de que no lo sean, se puede conseguir la unión-compatibilidad gracias a la proyección.

Operaciones de conjunto

- Mostrar todos los emails de las personas existentes en la base de datos

```
SELECT email  
FROM profesores
```

```
UNION
```

```
SELECT email  
FROM alumnos;
```

EMAIL
11087825@uma.es
11569523@uma.es
12127891@uma.es
13504134@uma.es
14156094@uma.es
14785585@uma.es
15940923@uma.es
16891547@uma.es
17647730@uma.es
20058694@uma.es
21012656@uma.es
22903027@uma.es
26651441@uma.es
27921947@uma.es
28179524@uma.es

...

Conservación de tuplas repetidas

- Las operaciones UNION, MINUS e INTERSECT eliminan tuplas repetidas.
- Ej: Mostrar todos los nombres de personas existentes en la base de datos, sin eliminar repeticiones.

```
SELECT nombre  
FROM profesores  
  
UNION ALL  
  
SELECT nombre  
FROM alumnos;
```

NOMBRE
Michael
Manuel
Enrique
Miguel
Maria del Mar
Sergio
Carlos
Ana
Miguel
Juana
Antonio
Guillermo
Manuel
Justo
Rafael

...

Producto Cartesiano

- Se puede utilizar en relaciones que no son Unión-compatibles

FROM *Lista_tablas*

- Se pueden utilizar alias de tabla:

FROM *tabla1* Alias1, *tabla2* alias2, ...

- Se utiliza la notación punto para identificar atributos con el mismo nombre en diferentes tablas (*tabla.atributo*)
- Tiene un **alto coste computacional**.

- Listar todas las combinaciones posibles de matriculación de todos los alumnos

```
SELECT AL.Nombre, ASI.Nombre  
FROM alumnos AL, asignaturas ASI;
```

NOMBRE	NOMBRE_1
Guillermo	Teoria de la señal
Manuel	Teoria de la señal
Justo	Teoria de la señal
Rafael	Teoria de la señal
Maria Jose	Teoria de la señal
Victoria	Teoria de la señal
Jose	Teoria de la señal
...	
Guillermo	Prácticas en empresa
Manuel	Prácticas en empresa
Justo	Prácticas en empresa
...	
Guillermo	Calculo Numerico
Manuel	Calculo Numerico
Justo	Calculo Numerico
Rafael	Calculo Numerico
...	

Selección

- La cláusula **WHERE** permite la seleccionar un subconjunto del cuerpo de una tabla basándose en un predicado

WHERE *Predicado*

- Toda fila que satisfaga el predicado aparecerá en el resultado de la consulta.
- Operadores relacionales: = , != , < , > , <= , >=
- Operadores booleanos: AND, OR, NOT
- Especiales: BETWEEN, LIKE

Selección

- Listar el Nombre de los profesores del departamento de código 1, ordenados alfabéticamente.

```
SELECT Nombre  
FROM profesores  
WHERE departamento = 1  
ORDER BY Nombre ;
```

⚡ NOMBRE
Manuel
Miguel
Sergio

- Listar el nombre y primer apellido de los profesores cuyo nombre empiece por 'M' y que pertenezcan a los departamentos con código entre 2 y 4

```
SELECT Nombre, Apellido1  
FROM profesores  
WHERE departamento BETWEEN 2 AND 4  
AND nombre LIKE 'M%';
```

⚡ NOMBRE	⚡ APELLIDO1
Michael	Brown
Maria del Mar	Roldán
Miguel	Hermoso

Funciones predefinidas

- Existen funciones predefinidas que se pueden utilizar tanto en el predicado de la cláusula WHERE como en la parte de proyección (SELECT ...)
- Cada tupla resultado de la consulta mostrará el resultado de aplicar la función utilizada al valor concreto para esa tupla de los atributos que se usan como parámetros de la función.
- http://docs.oracle.com/cd/B19306_01/server.102/b14200/functions001.htm#i88893

Funciones sobre cadenas de caracteres

- **CONCAT**(a,b) ó (a || b) concatena las cadenas *a* y *b*. Si *a* y *b* no son de tipo cadena se convierten antes de concatenarlos.
- **SUBSTR**(cad, pos, len) devuelve la subcadena de la cadena *cad*, que comienza en la posición *pos* y tiene longitud *len*.
 - La primera posición de la cadena es la 1.
 - Si pos = 0, entonces se considera que se desea empezar en 1.
 - Si pos < 0, la posición se determina empezando a contar desde el final de la cadena
- **LOWER**(cad) devuelve una cadena en minúsculas con los mismos símbolos que la cadena pasada.
- **UPPER**(cad) devuelve una cadena en mayúsculas con los mismos símbolos que la cadena pasada.
- etc

Funciones sobre cadenas de caracteres

- Mostrar un listado con una sola columna con alias 'Id' que muestre para cada profesor una cadena formada por las 3 primeras letras del nombre seguidas de las 3 últimas letras de su primer apellido.

```
SELECT substr(nombre,1,3) || substr(apellido1,-3,3) "Id"  
from profesores;
```

Id
Micown
Maniso
Enrler
Migtiz
Angora
Cesmez
Carssi
Ananez
Migoso
Juadez
Anteva

Funciones numéricas

- **Aritméticas:**
+, -, /, *, MOD, POWER,...
- **Trigonómicas:**
SIN, COS, TAN, ASIN, ACOS, ATAN,...
- **ROUND** (n,decimales) redondea n al número de decimales indicado
 - $\text{ROUND}(n) = \text{ROUND}(n,0)$
- **TRUNC** (n,decimales) trunca n al número de decimales indicado.
 - $\text{TRUNC}(n) = \text{TRUNC}(n,0)$
- etc.

Funciones numéricas

- Mostrar el resultado de multiplicar 3.1416 y 2.86, tal cual, redondeado y truncado a 3 decimales.

```
SELECT 3.1416*2.86, round(3.1416*2.86,3), trunc(3.1416*2.86,3)  
FROM dual;
```

3.1416*2.86	ROUND(3.1416*2.86,3)	TRUNC(3.1416*2.86,3)
8,984976	8,985	8,984

Funciones sobre fechas

- **ADD_MONTHS**(fechaInicio, m) devuelve la fecha resultante de sumar *m* meses a la *fechaInicio*.
- **MONTHS_BETWEEN** (fecha1, fecha2) devuelve la diferencia en meses entre *fecha1* y *fecha2*.
 - Si *fecha2* es más reciente que *fecha1* el número devuelto es negativo.
- *fecha1* – *fecha2*, devuelve la diferencia en días entre dos fechas.
 - Si *fecha2* es más reciente que *fecha1* el número devuelto es negativo.
- **NEXT_DAY** (fechaInicio, diaSemana) devuelve la fecha posterior a *fechaInicio* correspondiente al siguiente día que coincida con el día de la semana dado.
- **SYSDATE** devuelve la fecha actual del sistema.
- etc.

Funciones sobre fechas

- Devolver la fecha en que cae el siguiente miércoles a partir del día de hoy

```
SELECT NEXT_DAY(SYSDATE,'Miércoles')  
FROM DUAL;
```

- Mostrar el nombre de cada profesor junto a su edad.

```
SELECT nombre,  
       months_between(sysdate,fecha_nacimiento)/12 "Edad"  
FROM profesores
```

NOMBRE	Edad
Michael	74,78894016328156113102349661489446435683
Manuel	48,85076812027080844285145360414177618475
Enrique	43,87227349661489446435682994822779769017
Miguel	67,26743478693747510951812027080844285142
Angel	35,88033801274392672242134607726005575467
Cesar	42,59539177618478693747510951812027080842
Carlos	85,26743478693747510951812027080844285142
Ana	(null)
Miguel	(null)
Juana	24,14915521704500199123855037833532457192
Antonio	46,1437788729589804858622062923138191955

Funciones de conversión

- **TO_NUMBER** (cad) transforma la cadena de caracteres a número si el formato es el adecuado.
- **TO_CHAR** (param) convierte el número o fecha dada a cadena.
 - TO_CHAR(param,formato) realiza la conversión utilizando el formato dado
 - Se puede utilizar de TO_CHAR para extraer información del tipo fecha (sólo el mes, sólo el año, sólo el día de la semana, ...)
- **TO_DATE** (cad,formato) transforma una cadena a fecha siguiendo el formato dado.
- Para conocer cómo especificar el formato hay que consultar la referencia SQL
- http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm#i34510

Funciones de conversión

- Mostrar un listado con el nombre, el año y el día de la semana de nacimiento de cada profesor

```
SELECT      nombre,  
            TO_CHAR(fecha_nacimiento,'yyyy') "Año",  
            TO_CHAR(fecha_nacimiento,'DAY') "Día"  
FROM profesores;
```

NOMBRE	Año	Día
Michael	1942	SÁBADO
Manuel	1968	LUNES
Enrique	1972	DOMINGO
Miguel	1949	LUNES
Angel	1980	DOMINGO
Cesar	1974	MIÉRCOLES
Carlos	1931	SÁBADO
Ana	(null)	(null)
Miguel	(null)	(null)
Juana	1992	LUNES
Antonio	1970	MIÉRCOLES

Pseudo-columnas

- Toda tabla proporciona las pseudo-columnas ROWNUM y ROWID. Dichas columnas no están almacenadas en la tabla y sólo pueden consultarse.
- **ROWNUM**, contiene un identificador único para cada *tupla resultado* de la consulta. El identificador indica el orden en que fue seleccionada dicha tupla para la consulta.
- **ROWID**, contiene un identificador único para cada *fila* del *sistema*. Dicho identificador coincide con la localización en disco para encontrar dicha fila.

Pseudo-columnas

- Mostrar el nombre, rowid y rownum de sólo tres profesores.

```
SELECT nombre, rowid, rownum  
FROM profesores  
WHERE rownum < 4;
```

```
DESC profesores;
```

NOMBRE	ROWID	ROWNUM
Michael	AAASKrAAFAAAADDAAAA	1
Manuel	AAASKrAAFAAAADDAAB	2
Enrique	AAASKrAAFAAAADDAAC	3

Nombre	Nulo	Tipo
NRP	NOT NULL	VARCHAR2 (20)
NOMBRE	NOT NULL	VARCHAR2 (20)
APELLIDO1	NOT NULL	VARCHAR2 (20)
APELLIDO2		VARCHAR2 (20)
DEPARTAMENTO		NUMBER (3)
TELEFONO		VARCHAR2 (4)
EMAIL		VARCHAR2 (100)
DESPACHO		VARCHAR2 (10)
FECHA_NACIMIENTO		DATE
ANTIGUEDAD		DATE
DIRECTOR_TESIS		VARCHAR2 (20)

Ejercicios

- Mostrar:
 1. Grupo y calificación de todos los alumnos matriculados en la asignatura de código 112.
 2. Código de los profesores que no han dirigido tesis doctorales
 3. Código de los profesores que son directores de tesis y/o imparten asignaturas.
 4. Nombres de alumnos sin su inicial.
 5. Nombres de asignaturas que comiencen por 'B'.
 6. Dni de los alumnos que han aprobado alguna asignatura.
 7. Nombre -> fechaPrimMatrícula, para alumnos con nombre de más de 5 letras.
 8. Cantidad de meses desde la fecha de primera matrícula hasta el 1/12/2013.
 9. El nombre de una asignatura de menos de 6 créditos.