

FICHEROS y BB.DD.

# Práctica 1

## Estática Relacional

uc3m

Universidad  
**Carlos III**  
de Madrid

# Prácticas de la asignatura: hoja de ruta

## sesión 1

- Modelado Relacional (*esquema relacional*)
- Implementación: entorno SQL+ (consola interacción)
- Estática Relacional: creación de tablas en SQL (LDD)

práctica 1

## sesión 2

- Dinámica Relacional
  - consultas básicas en SQL y gestión transaccional
  - carga de datos (ejecución de scripts + volcado)
  - del álgebra relacional al SQL

práctica 2

## sesión 3

- Mecanismos de SQL avanzados
  - vistas y disparadores

## sesión 4

- Diseño Físico
  - Parametrización de la base
  - Organizaciones base y auxiliares
  - Hints

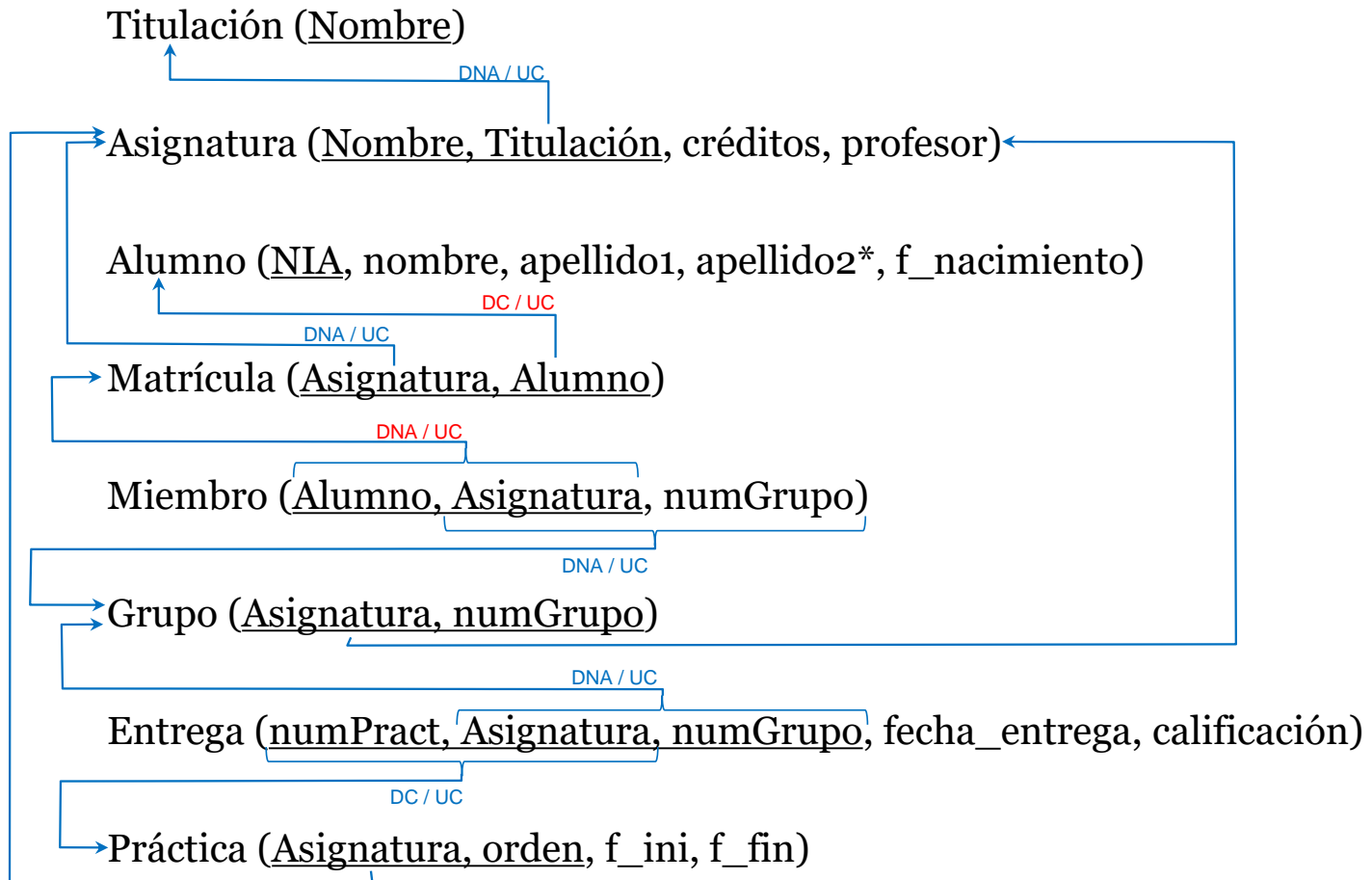
práctica 3

examen de prácticas

- Pasos a seguir:
  1. Identificación de relaciones
  2. Atributos dentro de la relación
  3. Atributos opcionales
  4. Interrelaciones, claves ajenas y restricciones de integridad
  5. Claves primarias y alternativas
  6. Descripción de la semántica explícita no contemplada
  7. Identificación de la semántica implícita imbuida
  8. Documentación (... ¡y a por la implementación!)

- Se necesita una base de datos para gestionar las prácticas:
  1. Debe recogerse la información de los alumnos (NIA, nombre, primer apellido y opcionalmente segundo apellido, edad y fecha de nacimiento)
  2. También hay asignaturas: nombre, créditos, titulación, profesor
  3. Debe recogerse la formación de grupos de prácticas. Los grupos tienen un identificador numérico (3) distinto para cada asignatura, y los forman alumnos matriculados en la asignatura.
  4. Las prácticas de cada asignatura se identifican por su núm\_orden y tienen enunciado, plazo (fecha inicio, fecha fin), y calificación para cada grupo.
  5. Un alumno no puede cancelar matrícula mientras sea miembro de un grupo. Si un alumno es expulsado (eliminado de la base) se cancelan todas sus matrículas. Si una práctica desaparece, también lo hacen sus entregas. En el resto de los vínculos, no deben permitirse borrados que dejen registros huérfanos. Todas las modificaciones se propagan.

# Diseño Relacional: Ejemplo



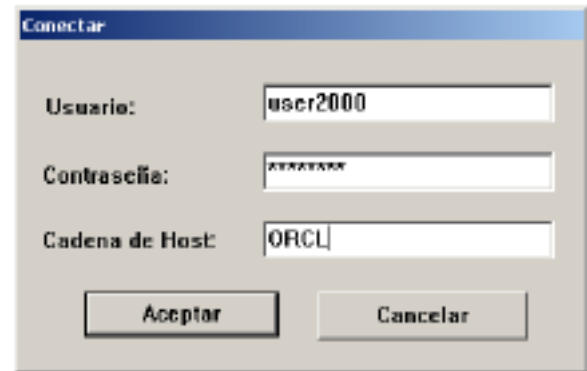
- Herramienta interfaz básica de Oracle
  - Permite conectarse al Servidor Oracle, y enviarle instrucciones SQL para su ejecución (mostrando después los resultados).
  - Dispone de su propio lenguaje de *comandos*
  - Permite la ejecución de *scripts SQL*
    - ✦ *script: fichero de texto que contiene un conjunto de instrucciones*
    - ✦ *ejemplo de script SQL: fichero “creacion.sql”*

- Para iniciar la consola:

**sqlplus** [usuario[/password[@sid]]] [@script [par1 [...]]]

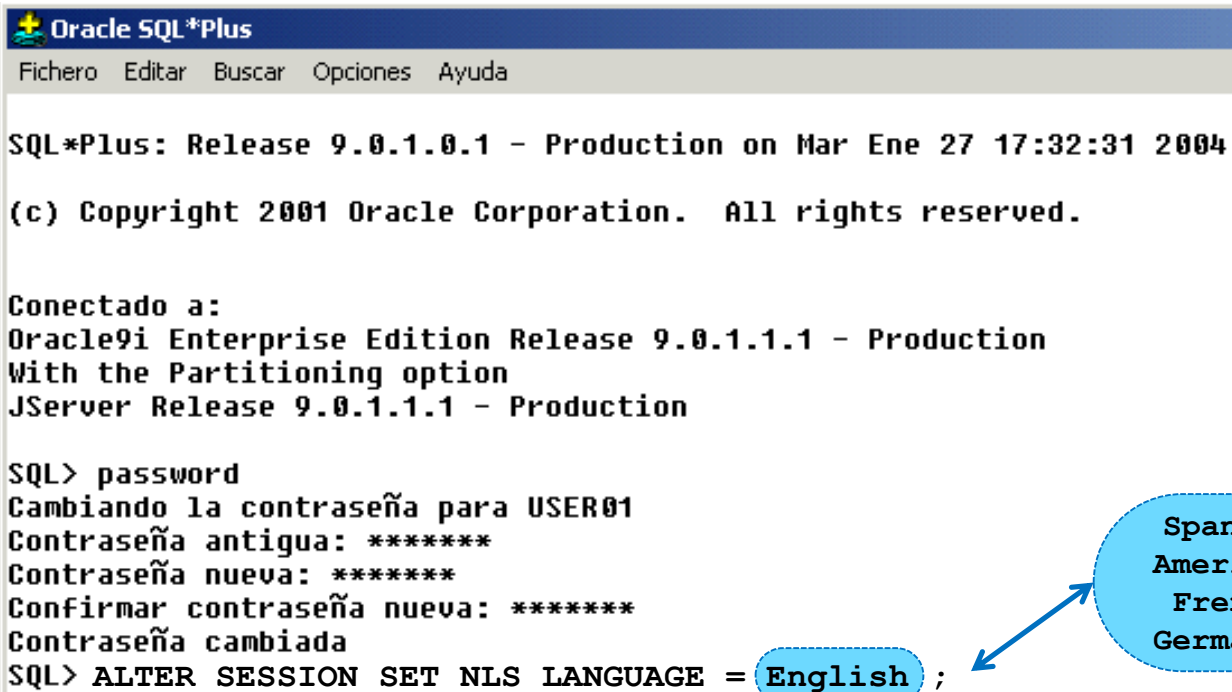
- Si se omiten usuario y/o password, los solicitará al iniciar
- Si se omite la cadena de conexión, tomará la instancia por defecto
- Si se especifica script pero no usuario (o se omite el password), esta(s) deberán ser las primeras líneas del script
- Para iniciar sin conectarse a una bd, `sqlplus /nolog`

- Para trabajar con una instancia de BD (mediante SQL+ o cualquier interfaz), es necesario conectarse
- En la conexión, SQL+ solicitará al usuario:
  - Nombre de usuario
  - Contraseña
  - Cadena de conexión (SID) (identificación del servicio)
- En este momento, el profesor dará una cuenta en ORACLE a cada grupo



# SQL+: Primeros comandos

- Básicos: `connect`, `disconnect`, `help`, `!comando_SO`, etc.
- Primer paso: **cambiar la contraseña** con `password`



```
Oracle SQL*Plus
Fichero  Editar  Buscar  Opciones  Ayuda

SQL*Plus: Release 9.0.1.0.1 - Production on Mar Ene 27 17:32:31 2004

(c) Copyright 2001 Oracle Corporation.  All rights reserved.

Conectado a:
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

SQL> password
Cambiando la contraseña para USER01
Contraseña antigua: *****
Contraseña nueva: *****
Confirmar contraseña nueva: *****
Contraseña cambiada
SQL> ALTER SESSION SET NLS_LANGUAGE = English ;
```

Spanish,  
American,  
French,  
German, ...



# SQL+: configurar el entorno

- Mostrar variables de entorno, parámetros *pfile*, errores...

```
show var_entorno
```

```
show parameters [filter]
```

```
show errors [element_type [element_name]]
```

- Asignar un valor a una variable

```
set var_entorno valor
```

- Variables entorno importantes:

```
set wrap off          /* desactiva ajuste de línea */
```

```
set linesize 900      /* cambia long línea (default 80) */
```

```
Otras (on/off): pause, timing, heading, echo,  
serveroutput, autocommit, autotrace,...
```

```
Otras: pagesize, sqlprompt, null, ...
```

- Guardar configuración actual: `store set <file> replace`  
(el fichero de configuración del entorno se carga con @<file>)

- Limpia la pantalla / el buffer: `clear screen | buffer`
- Lista el contenido del buffer: `L`
- Ejecuta el buffer : `r`
- Cambia la línea actual: `1 | 2 | 3 | ...`
- Añade un texto a la línea actual: `a text`
- Añade una línea tras la línea actual: `i text`
- Cambia un texto por otro la línea actual  
`c/original/nuevo`    */\* si se omite nuevo, borra \*/*
- Cierra el modo de edición de línea: `/`

# SQL+: buffer , scripts y spooling

- Editar el buffer: `ed`
- Cambiar el nombre/ruta del fichero de buffer:  
`set editfile ruta\filename /* privilegio escritura */`
- Editar un script: `ed <fichero>`
- Ejecución de un script:  
`start <fichero> | @<fichero> | @@<fichero>`
- Cargar un fichero en el buffer: `get <fichero>`
- Guardar el contenido del buffer: `save <fichero>`
- Guardar la salida en un fichero (spooling):  
`spool <fichero> [cre|rep|app]`
- Consultar estado del spooling [detenerlo]: `spool [OFF]`

# SQL+: variables de usuario (scripts)

- Visualización/asignación de variable usuario

```
define          /* muestra las variables definidas */  
define var      /* muestra el valor de var */  
define var=valor /* asigna el valor a var */
```

- valor es texto; puede contener hasta 240 caracteres
- si *valor* contiene ' deberá entrecomillarse: `var="'...'"`
- las variables predefinidas empiezan con `_`, como `_user`
- cambia editor por defecto: `define _editor=notepad.exe`

- Solicitud de un valor para una variable (al usuario)

```
accept [number] [prompt text] var
```

- es implícito si se utiliza una var que no está definida

- Suprimir una variable: `undefine var`

# SQL+: variables de usuario (II)

- Las variables de usuario son variables de sustitución (sql+ sustituye la referencia por su valor **antes** de ejecutar la instrucción)
- Uso de variables (especialmente útiles en scripts)
  - `&variable`    /\* se sustituye por su valor al ejecutar \*/
  - `&variable.`   /\* igual, pero sin añadir espacio detrás \*/
  - `&&variable`   /\* igual, pero usa y define \*/
- Parámetros del script: son valores de texto separados por espacios tras la llamada @fichero
- Los parámetros se referencian con las variables `&1`, `&2`, `&3`, ...
- Variables entorno importantes:
  - `set define off`        /\* desactiva variables en scripts \*/
  - `set verify off`       /\* desactiva eco de la sustitución \*/

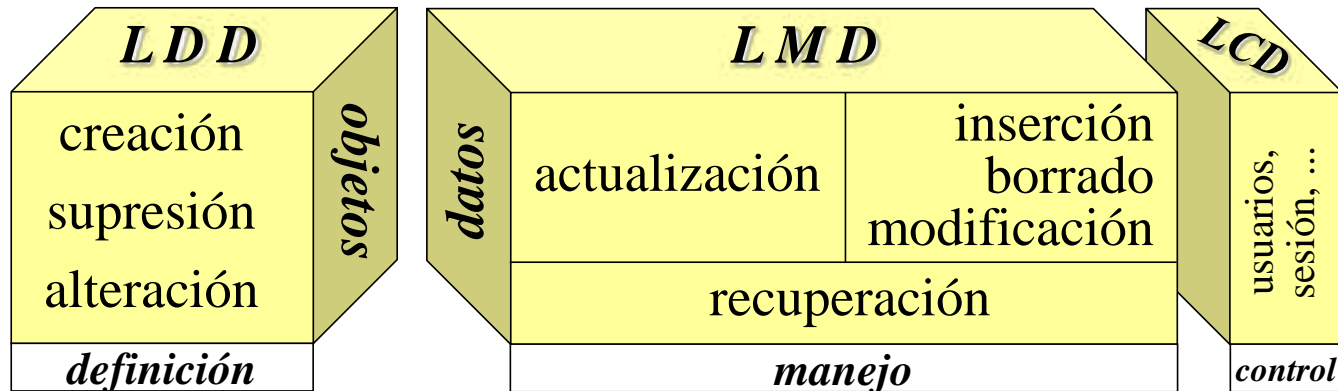
# SQL+: otras utilidades (en scripts)

- Comentarios multilínea: `/* ..... */`
- Comentarios de una línea: `remark | --`
- Muestra un mensaje (puede contener variables): `prompt ...`
- Pausa hasta que se pulse intro: `pause [text]`
- Termina el script (exit/quit):  
`exit [success|failure|warning|n] [commit|rollback]`
- Variables ligadas: son accesibles desde SQL  
`variable varname [type]`
- Asignar una variable ligada: `:varname := value`  
(también desde SQL, con `select col into :varname from ... ;`)
- Visualizar una variable ligada: `print :varname`

# SQL $\neq$ SQL\*plus

- Instrucciones SQL: terminan siempre en ;
  - Consulta de tablas de usuario: (instrucción SQL)  
`select * from user_tables where table_name='...';`
  - Descripción de un objeto: (comando SQL+)  
`desc <table>`
- Se ejecutan en el servidor (no en local)
- No diferencia mayúsculas de minúsculas (sí en *strings*)
- Los *strings* se encierran con comillas simples: ' ... '  
(si el *string* contiene comilla simple debe duplicarse: 'Let' 's go')
- Otros datos, como las fechas, requieren conversión

- Instrucciones del Lenguaje SQL: tres tipos
  - Lenguaje de Definición de Datos
  - Lenguaje de Manipulación de Datos
  - Lenguaje de Control de Datos





- Tres acciones básicas:
  - Creación (CREATE)
  - Supresión (DROP)
  - Alteración (ALTER)
- Sobre:
  - Esquemas
  - Tablas
  - Vistas
  - Dominios
  - Secuencias...

**Creación\_de\_tabla** :=

```
CREATE [GLOBAL TEMPORARY] TABLE nombre_tabla
{ (elemento_de_tabla [, elemento_de_tabla [, ...] ] )
| [AS subquery] }
[opciones_de_almacenamiento]
;
```

**elemento\_de\_tabla** :=

```
{ nombre_columna datatype(maxsize) [column_constraints]
| restricción_tabla }
```

```
[ON COMMIT {DELETE|PRESERVE} ROWS]
```

# Creación en LDD: crear una tabla (I)

```
CREATE [GLOBAL TEMPORARY] TABLE nombre_tabla
{ (elemento_de_tabla [, elemento_de_tabla [,...]] )
  | AS (<subquery>) [WITH NO DATA] }
[opciones_de_almacenamiento]
;
```

*elemento\_de\_tabla* := { *def\_columna* | *restr\_tabla* }

*def\_columna* := *nombre\_columna* **datatype**(**max\_size**)  
[*val\_column*]  
[*restricción\_columna*]

*val\_column* := { **DEFAULT** *valor*  
| **GENERATED** {**ALWAYS**|**BY DEFAULT**} **AS IDENTITY**  
[ (**START WITH** *int* [, **INCREMENT BY** *int*] ) ] }

# Creación en LDD: crear una tabla (II)

## RESTRICCIONES:

- de columna

[CONSTRAINT nombre ]

[NOT] NULL  
PRIMARY KEY  
UNIQUE  
REFERENCES...  
CHECK ( ... )

[estado]

ENABLE  
DISABLE  
...

- de tabla

[CONSTRAINT nombre ]

PRIMARY KEY (col[,...])  
UNIQUE (col[,...])  
FOREIGN KEY (col[,...]) REFERENCES ...  
CHECK ( ... )

[estado]

- Interrelaciones:

Clave ajena: como una restricción más (a nivel de columna o de tabla)

```
FOREIGN KEY (<columnas>)
    REFERENCES <nombre_tabla>
    [(<columnas>)]
    [ON DELETE <acción>]
    [ON UPDATE <acción>]
```

Acciones referenciales (Reglas de Integridad) definidas en el estándar

**CASCADE | SET NULL | RESTRICT | NOT ACTION | SET DEFAULT**

- Oracle permite las siguientes:
  - Actualización: sólo implementa la **NO ACTION**
  - Borrado: por omisión **NO ACTION**; soporta **CASCADE** y **SET NULL**

# LDD para crear tablas: EJEMPLOS

```
CREATE TABLE ejemplo(  
    identifiier NUMBER(3) PRIMARY KEY,  
    titulo VARCHAR2(50) NOT NULL,  
    cantidad NUMBER(8) DEFAULT(8),  
    CONSTRAINT ttl_valido CHECK titulo IN ('A','B','C')  
);
```

DESC ejemplo

/\* Podemos encontrar información más completa en el catálogo \*/

```
SELECT table_name FROM USER_TABLES;
```


```
SELECT column_name, data_type, data_length, nullable  
    FROM USER_TAB_COLUMNS  
    WHERE table_name='ejemplo';
```

```
/* Al catálogo relacional Oracle lo denomina 'Data Dictionary' */  
/* http://docs.oracle.com/cd/B19306\_01/server.102/b14237/statviews\_2105.htm */  
/* http://docs.oracle.com/cd/B19306\_01/server.102/b14237/statviews\_2094.htm */  
/* http://docs.oracle.com/cd/B10501\_01/server.920/a96536/ch229.htm */
```

# LDD para crear tablas: EJEMPLOS

```
CREATE TABLE referenciante(  
    nombre VARCHAR2(50),  
    fecha DATE DEFAULT SYSDATE,  
    padre NUMBER(3) CONSTRAINT primogenito UNIQUE,  
    CONSTRAINT PK_referenciante PRIMARY KEY (nombre),  
    CONSTRAINT FK_ref2ejemplo FOREIGN KEY (padre)  
        REFERENCES ejemplo(identifier) ON DELETE CASCADE  
);
```

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE  
FROM USER_CONSTRAINTS  
WHERE table_name='referenciante';
```



C	Check
P	Primary K
U	Unique
R	Referential
V	view's CO
O	view's RO

```
SELECT CONSTRAINT_NAME, R_CONSTRAINT_NAME, DELETE_RULE  
FROM USER_CONSTRAINTS  
WHERE table_name='referenciante' AND CONSTRAINT_TYPE='R';
```

```
SELECT CONSTRAINT_TYPE, TABLE_NAME  
FROM USER_CONSTRAINTS WHERE constraint_name='...';
```

## 2. Supresión:

```
DROP <elemento> <nombre>;
```

```
TRUNCATE <elemento> <nombre>;
```

## 3. Alteración:

```
ALTER <elemento> <nombre>  
    {ADD|ALTER|DROP} <elemento>  
    [<definición>;
```

### EJEMPLOS:

```
ALTER TABLE ejemplo ADD (edad NUMBER(2));
```

```
ALTER TABLE ejemplo ADD CONSTRAINT max_edad CHECK (edad<100);
```

```
SELECT CONSTRAINT_NAME, SEARCH_CONDITION  
    FROM USER_CONSTRAINTS  
    WHERE table_name='ejemplo' AND CONSTRAINT_TYPE='C';
```

```
DROP TABLE persona;
```



# Ejercicio: Creación de tablas

- Crear las tablas del ejercicio de clase (BD prácticas, diap 5).
  1. Creadlas en la consola, y cuando la instrucción sea satisfactoria, pasadlas a un documento de texto (script) donde se recogerán todas.
  2. Empezad por titulación y alumno, y seguid luego con asignatura, matrícula, grupo, miembro, práctica y entrega. Observad que el orden es importante.
  3. Una vez creadas todas las tablas en el script, dadle robustez incluyendo en su cabecera una destrucción de tabla por cada tabla creada después.
  4. Ejecutad el script para verificar su funcionamiento.
  5. Dadle ahora más semántica añadiendo restricciones: dominios (calificación de 0 a 10; créditos de 1 a 60), fechas pasadas (nacimiento, entrega), lapsos correctos (nada puede terminar antes de empezar), etc.
  6. Ejecutad y comprobad en el catálogo.