

# Prácticas EDAT

## Curso 11/12

Alejandro Bellogín

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Septiembre 2011

<http://www.eps.uam.es/~abellogin>

# Esquema

- Parte I
  - Contacto
  - Organización de las prácticas
    - Normas
    - Calendario
- Parte II
  - Nociones básicas de SQL
  - Comandos básicos de SQL
  - Comandos avanzados de SQL
- Parte III
  - Explicación de la práctica
  - A practicar

# Esquema

- Parte I
  - Contacto
  - Organización de las prácticas
    - Normas
    - Calendario
- Parte II
  - Nociones básicas de SQL
  - Comandos básicos de SQL
  - Comandos avanzados de SQL
- Parte III
  - Explicación de la práctica
  - A practicar

# Contacto

- **Correo electrónico** (preferible: asunto '[edat]'):  
alejandro . bellogin @ uam . es
- Despacho: B – 408
- Hora de tutorías? ← acordadas previamente!
  - Tentativo: Jueves de 18:00 a 20:00
- En clase:
  - miércoles de 18:00 a 20:00
  - viernes de 18:00 a 20:00
- Teléfono: 91 497 22 93

# Normativa

- **Formación de parejas y asignación de grupos a prácticas**
  - Trabajo en parejas, asistiendo al mismo horario (grupo de prácticas)
  - Asignación de una pareja a un grupo de prácticas realizada por los profesores atendiendo a formularios de petición (recogidos en las primeras clases de teoría)
- **Asistencia a clase**
  - Obligatoria
  - Control de la puntualidad
- **Copia de prácticas**
  - Sanción según el Capítulo V de la EPS: apertura de expediente de expulsión o suspenso automático sin convocatoria extraordinaria
  - Copiador y copiado

# Normativa

- **Calendario y retrasos de entregas de prácticas**
  - 4 prácticas
    - Práctica 1: Introducción a bases de datos relacionales y PostgreSQL (1 semana)
    - Práctica 2: Consultas a bases de datos relacionales (3 semanas)
    - Práctica 3: Diseño de bases de datos relacionales (4 semanas)
    - Práctica 4: Implementación de bases de datos – Indexación (5 semanas)
  - Calificación final: 10% P1 + 25% P2 + 30% P3 + 35% P4
  - Calificación mínima en cada práctica: 3
  - Fechas de entrega en los enunciados de las prácticas y en Moodle
  - Entrega de una práctica hasta las 12 de la noche del día anterior a la fecha límite correspondiente
  - Penalizaciones en la calificación por retraso en la entrega
    - Por el primer día de retraso: – 20% de la calificación de la práctica
    - Por cada día siguiente de retraso: – 10% de la calificación de la práctica

# Normativa

- **Forma de entrega y contenidos de los ficheros entregados**
  - Envío vía Moodle
  - Entrega de 1 fichero **edat1112\_pX\_YYY\_ZZ.zip**
    - X = Número de práctica (1, 2, 3 ó 4)
    - YYY = Grupo de prácticas (211, 212, 214, 261, 262, 271, 272)
    - ZZ = Número de pareja
    - Ejemplo: práctica 3, pareja 1 del grupo 214 → edat1112\_p3\_214\_01.zip
  - El contenido del .zip dependerá de cada práctica y se detallará en el enunciado de la misma
  - Penalización en la calificación por formato incorrecto: 0.5 puntos
- **Corrección de las prácticas y publicación de calificaciones**
  - Inspección del material entregado y posibles preguntas personalizadas en clase de prácticas
  - Publicación de calificaciones en Moodle

# Calendario

- **23 septiembre** - presentación, explicación P1
- 30 septiembre - entrega y corrección P1, explicación P2
- 07 octubre - continuación P2
- 14 octubre - continuación P2
- **21 octubre - entrega y corrección P2, explicación P3**
- **28 octubre - continuación P3**
- 04 noviembre - continuación P3 + **corrección P2**
- 11 noviembre - continuación P3
- 18 noviembre - entrega y corrección P3, explicación P4
- 25 noviembre - continuación P4
- 02 diciembre - continuación P4
- 09 diciembre - continuación P4
- 16 diciembre - entrega y corrección P4



# Esquema

- Parte I
  - Contacto
  - Organización de las prácticas
    - Normas
    - Calendario
- Parte II
  - Nociones básicas de SQL
  - Comandos básicos de SQL
  - Comandos avanzados de SQL
- Parte III
  - Explicación de la práctica
  - A practicar

# Nociones básicas de SQL (I)

- SQL es un lenguaje de programación de SGBDs (software de gestión y acceso a BDs)
- Permite
  - Crear modelos de datos (diseño de tablas) (DDL)
  - Insertar/modificar/eliminar datos (DML)
  - Consultar datos (DQL)
- Se ejecuta sobre un SGBD
- Hay diferencias entre implementaciones → aprender 'variantes' de SQL
  - En este curso: postgresSQL
  - Otros: MySQL, Oracle, ...

# Nociones básicas de SQL (II)

- Base de datos (BD) = conjunto de tablas
- Tabla (relación, entidad esquema) =
  - Estructura fija de campos (esquema)
  - Registros con valores de campos (datos)
- Campo (atributo, propiedad, *columna*)  
con un tipo de dato determinado
- Registro (tupla, *fila*)
- Clave primaria = identificador único
- Clave externa = identificador de otra tabla

# Nociones básicas de SQL – Ejemplo (I)

- **Base de datos relacional**
  - **Tablas** (asociadas a “entidades” sobre las que guardamos datos)
    - **Columnas:** atributos de la entidad
    - **Filas:** instancias particulares de la entidad

## EMPLEADO

NUM_EMP	NOMBRE	TARIFA_HORA	CUALIF.	NUM_SUP
2	C. García	16	Fontanero	5
3	R. Gómez	30	Carpintero	4
6	C. Barrero	16	Electricista	4

## ASIGNACION

NUM_EMP	NUM_ED	FECHA_INI	NUM_DIAS
2	1	20/12	5
3	2	28/10	10
6	4	17/11	12
2	2	15/10	15
3	1	01/08	22
6	2	08/10	12

# Nociones básicas de SQL – Ejemplo (II)

- Esquema relacional

- **Tablas** = tupla de atributos

- **Claves primarias** (*primary keys*): atributos que identifican de forma unívoca a instancias de la tabla (entidad) [equivalente a UNIQUE + NOT NULL]
- **Claves externas/foráneas** (*foreign keys*): atributos que se corresponden con claves primarias de otras tablas

NUM_EMP	NOMBRE	TARIFA_HORA	CUALIF.	NUM_SUP
2	C. García	16	Fontanero	5
3	R. Gómez	30	Carpintero	4
6	C. Barrero	16	Electricista	4

NUM_EMP	NUM_EDICION	FECHA_INI	NUM_DIAS
2	1	20/12	5
3	2	28/10	10
6	4	17/11	12
2	2	15/10	15
3	1	01/08	22
6	2	08/10	12

# Nociones básicas de SQL – Ejemplo (III)

- **Esquema relacional**
  - **Tablas** = tuplas de atributos

EMPLEADO (num\_emp, nombre, tarifa\_hora, cualificacion, num\_sup)

EDIFICIO (num\_edi, direccion, tipo, nivel, estado)

ASIGNACION (num\_emp, num\_edi, fecha\_ini, num\_dias)

# Comandos básicos de SQL

- Consultas (con restricciones/condiciones)
- Producto cartesiano vs Join
- Álgebra de conjuntos
- Ordenar resultados

# Comandos básicos de SQL – Ejemplo (I)

- **Consultas a una sola tabla**

- Ejemplo: Obtener el nombre de todos los fontaneros

```
SELECT NOMBRE  
FROM EMPLEADO  
WHERE CUALIFICACION= 'Fontanero'
```

```
SELECT columnas  
FROM tabla  
WHERE condición
```



# Comandos básicos de SQL – Ejemplo (II)

- Consultas a una sola tabla

- Ejemplo: Obtener el nombre de todos los fontaneros

NUM_EMP	NOMBRE	TARIFA_HORA	CUALIF.	NUM_SUP
1	M. Pérez	12	Electricista	5
2	C. García	16	Fontanero	5
3	R. Gómez	30	Carpintero	4
4	P. Morán	16	Pintor	4
5	H. Rico	18	Fontanero	5
6	C. Barrero	16	Electricista	4
7	A. Cerrato	30	Pintor	4

resultado  
consulta



<b>NOMBRE</b>
C. García
H. Rico

# Comandos básicos de SQL – Ejemplo (III)

- **Consultas a una sola tabla**

- Ejemplo: Listar todos los datos sobre edificios de oficinas

```
SELECT *  
FROM EDIFICIO  
WHERE TIPO= 'Oficina'
```

# Comandos básicos de SQL – Ejemplo (IV)

- Consultas a una sola tabla

- Ejemplo: Listar todos los datos sobre edificios de oficinas

NUM_EDI	DIRECCION	TIPO	NIVEL	ESTADO
1	Azcoitia, 12	Oficina	2	2
2	Pelícano, 89	Tienda	1	1
3	Argüeso, 24	Vivienda	3	1
4	Mineros, 2	Oficina	3	1
5	Tavira, 78	Oficina	4	1
6	Alcalá, 59	Almacén	3	3

resultado  
consulta



NUM_EDI	DIRECCION	TIPO	NIVEL	ESTADO
1	Azcoitia, 12	Oficina	2	2
4	Mineros, 2	Oficina	3	1
5	Tavira, 78	Oficina	4	1

# Comandos básicos de SQL – Ejemplo (V)

- **Los operadores de comparación**
  - Ejemplo: ¿Quién tiene una tarifa por hora entre 10 y 20 Euros?

```
SELECT *  
FROM EMPLEADO  
WHERE TARIFA_HORA >= 10  
      AND TARIFA_HORA <= 20
```

Operadores de comparación: =, <>, <, >, <=, >=

Conectivas booleanas: AND, OR, NOT

# Comandos básicos de SQL – Ejemplo (VI)

- **El operador IN**

- Ejemplo: Listar todos los datos sobre fontaneros, carpinteros y electricistas

```
SELECT *  
FROM EMPLEADO  
WHERE CUALIFICACION IN ('Fontanero',  
    'Carpintero', 'Electricista')
```

# Comandos básicos de SQL – Ejemplo (VII)

- Consultas a varias tablas
  - Producto cartesiano

```
SELECT nombre, num_dias  
FROM EMPLEADO, ASIGNACION  
WHERE EMPLEADO.num_emp = ASIGNACION.num_emp
```

- Join

```
SELECT nombre, num_dias  
FROM EMPLEADO JOIN ASIGNACION ON num_emp
```

# Comandos básicos de SQL – Ejemplo (VIII)

- **Álgebra de conjuntos**
  - Usado con consultas anidadas

```
(SELECT nombre FROM EMPLEADO WHERE num_emp<2)  
UNION  
(SELECT nombre FROM EMPLEADO WHERE num_emp>6)
```

- INTERSECT, EXCEPT
- Por defecto, elimina duplicados, salvo que se indique ALL

# Comandos básicos de SQL – Ejemplo (IX)

- Ordenar resultados

```
SELECT nombre  
FROM EMPLEADO  
WHERE CUALIFICACION = 'Fontanero'  
ORDER BY nombre DESC
```

<b>NOMBRE</b>
C. García
H. Rico



<b>NOMBRE</b>
H. Rico
C. García



# Comandos avanzados de SQL

- Agregar resultados y ejecutar funciones
- Alias – ‘as’
- Vistas
- Obtener resultados sin duplicados (“distintos”)
- Operaciones con cadenas

# Comandos avanzados de SQL – Ejemplo (I)

- **Agregar resultados y ejecutar funciones**
  - Se pueden agrupar filas de acuerdo a un campo

```
SELECT cualificacion, count(*)  
FROM EMPLEADO  
GROUP BY cualificacion
```

CUALIFICACION	COUNT(*)
ELECTRICISTA	2
FONTANERO	2
CARPINTERO	1
PINTOR	2

- Y se aplican funciones sobre los resultados de la agrupación:

- COUNT
- SUM
- MAX
- MIN
- AVG

CUALIF.	NUM_SUP
Electricista	5
Fontanero	5
Carpintero	4
Pintor	4
Fontanero	5
Electricista	4
Pintor	4

# Comandos avanzados de SQL – Ejemplo (II)

- **Alias**

- Se aplican a tablas y atributos

```
SELECT nombre, num_dias  
FROM EMPLEADO AS EM, ASIGNACION AS A  
WHERE EM.num_emp = A.num_emp
```

```
SELECT cualificacion, count(*) AS c  
FROM EMPLEADO  
GROUP BY cualificacion  
ORDER BY c
```

# Comandos avanzados de SQL – Ejemplo (III)

- **Vistas**

- Dan nombre a una consulta
- Útil para reutilizar

```
CREATE VIEW font_pint_nombres AS
SELECT nombre, tarifa_hora, num_sup, direccion
FROM empleado AS em, asignacion AS a, edificio AS ed
WHERE em.num_emp = a.num_emp AND
      ed.num_edi = a.num_edi AND
      CUALIFICACION IN ('Fontanero', 'Pintor');
```

```
SELECT nombre
FROM font_pint_nombres
WHERE tarifa_hora < 17;
```

# Comandos avanzados de SQL – Ejemplo (IV)

- **Obtener resultados sin duplicados**

- Se usa el operador DISTINCT sobre el campo determinado
- Ejemplo: Listar todos los valores de tarifa por hora distintos

```
SELECT DISTINCT TARIFA_HORA  
FROM EMPLEADO
```

NUM_EMP	NOMBRE	TARIFA_HORA	CUALIF.	NUM_SUP
1	M. Pérez	12	Electricista	5
2	C. García	16	Fontanero	5
3	R. Gómez	30	Carpintero	4
4	P. Morán	16	Pintor	4
5	H. Rico	18	Fontanero	5
6	C. Barrero	16	Electricista	4
7	A. Cerrato	30	Pintor	4

TARIFA_HORA
12
16
30
18

# Comandos avanzados de SQL – Ejemplo (V)

- **Operaciones con cadenas**
- El operador LIKE
  - Ejemplo: Listar todos los datos sobre empleados cuya cualificación empieza por “Elec”

```
SELECT *  
FROM EMPLEADO  
WHERE CUALIFICACION LIKE 'Elec%'
```

% – cero o más caracteres

\_ – un carácter

- Concatenación

```
CONCAT(nombre, num_dias)
```

- Expresiones regulares

```
SIMILAR TO '%(lec)|(ont)%'
```

# Esquema

- Parte I
  - Contacto
  - Organización de las prácticas
    - Normas
    - Calendario
- Parte II
  - Nociones básicas de SQL
  - Comandos básicos de SQL
  - Comandos avanzados de SQL
- Parte III
  - Explicación de la práctica
  - A practicar

# Práctica 1

- Crear tablas que representen los datos entregados
  - *Filmoteca: películas, actores, directores*
- ¿Claves primarias/externas? ¿Tabla(s) adicionales?
- Insertar datos
- Ejecutar consultas
  
- Entregar:
  - Respuestas al uso de claves y tablas adicionales
  - *Dump* de la base de datos
  - Fichero con las consultas



# Practica: 'Hola Mundo' en SQL

- Crear base de datos
- Arrancar el programa de postgres
- Crear una tabla
- Insertar datos
- Ejecutar consulta

# Practica (I)

- **Crear base de datos**

- `createdb -U usuario bd`

- En los laboratorios (Linux):

- usuario = 'alumnodb'

- password = (ninguna)

- Para el ejemplo:

- db = 'prueba'

# Practica (II)

- **Arrancar el programa de postgres**
  - `psql -U alumnodb prueba`

# Practica (III)

- **Crear una tabla**

- Sentencia CREATE TABLE

```
CREATE TABLE nombre  
(definición_de_columnas,  
restricciones_de_tabla)
```

```
CREATE TABLE tabla (  
    s varchar(64)  
);
```

# Practica (IV)

- **Insertar datos**

- Sentencia INSERT

```
INSERT INTO nombre_tabla(nombres_columnas)  
VALUES (valores_columnas)
```

```
INSERT INTO tabla  
VALUES ('Hola'), ('mundo'), ('!');
```

```
INSERT INTO tabla VALUES ('Hola');  
INSERT INTO tabla VALUES ('mundo');  
INSERT INTO tabla VALUES ('!');
```

# Practica (V)

- **Ejecutar consulta**

```
SELECT *  
FROM tabla;
```

- **Salir: \q**

# Más cosas de postgresSQL

- Más tipos de datos
- Volcado de base de datos
- Carga de una base de datos
- Eliminar base de datos
- Vaciar tabla (sólo datos)
- Eliminar tabla (datos + estructura)
- Modificar tabla

# Más cosas de postgresQL

- **Tipos de datos (algunos)**
  - **char(n)**
    - Cadena de caracteres de longitud fija
  - **varchar(n)**
    - Cadena de caracteres de longitud variable
  - **integer**
    - Enteros, 4 Bytes, rango: -2147483648 a +2147483647
  - **real / double precision**
    - Reales, 4 Bytes / 8 Bytes, 6 decimales / 15 decimales
  - **serial**
    - Enteros auto-incrementales ¡Muy útiles como claves primarias!
  - **boolean**
    - TRUE, 't', 'true', 'y', 'yes', '1', FALSE, 'f', 'false', 'n', 'no', '0'
  - **date / time**
    - Fecha / hora



# Más cosas de postgresQL

- Volcado de base de datos

```
pg_dump --inserts -U usuario bd > fichero  
pg_dump --inserts -a -U usuario bd >  
fichero
```

- Carga de una base de datos

```
psql -U usuario -d bd -f fichero
```

- Eliminar base de datos

```
dropdb -U usuario bd
```

# Más cosas de postgresQL

- Copiar datos de una tabla

`=# copy tabla from fichero`

`=# copy tabla to fichero`

*'=#' representa que se invoca desde la consola de pSQL*

- Vaciar tabla (sólo datos)

`=# truncate table tabla`

`=# delete from tabla`

- Eliminar tabla (datos + estructura)

`=# drop table tabla`

# Más cosas de postgresQL

- Modificar tabla

```
=# alter table tabla add/drop/modify ..
```

FIN