

# FUENTES DE DOCUMENTACIÓN

- [www.postgresql.org](http://www.postgresql.org)
  - Para cada versión: Documentación estática o en línea
- manuales de alguna distribución: carpeta docs/html iniciando en “index.html”
  - “Tutorial”
  - Guía de usuario
  - Manual de referencia
  - Guía de programación
  - Guía del administrador

# TIPOS DE OBJETOS

- NUMÉRICOS
  - INTEGER..., NUMERIC,.. (FLOAT): REAL, DOUBLE...
- CARACTERES
  - CHARACTER (VARYING),,, TEXT
- FECHA-HORA
- BOOLEANOS
- GEOMÉTRICOS

# FUNCIONES Y OPERADORES

- OPERADORES LÓGICOS (AND OR NOT)
- COMPARACIONES (< <= = > >= <> !=)
- MATEMÁTICOS (+ - \* / % ^ ||/ ! !! @ & | # ...)
  - FUNCIONES (pi(), pow(p, n), sin(a), ....)
- CADENAS DE CARACTERES
  - substr(sb,st[,n]), position( a in b) trim.(l r b), translate
- CASAR PATRONES (sql: texto LIKE patron%)
  - regex: ~ ~\* !~ !~\* ... a | (b | c) . ? \* + []
- GEOMETRICOS ...

# AGREGADOS (group by)

- COUNT
- MAX, MIN
- SUM
- AVG
- STDDEV
- VARIANCE
- pueden programarse agregados nuevos

# FUNCIONES DE SUBCONSULTA

- EXISTS ( ... )
- e IN ( ... )      e NOT IN ( ... )
  - e = ANY ( ... )
- e op ANY ( ... )
- e op ALL ( ... )
  - op es cualquier operador entre e y los resultados del select entre paréntesis

# ÓRDENES SQL BÁSICAS

- CREATE TABLE
  - CREATE INDEX
  - ALTER (DROP) TABLE
  - GRANT (REVOKE)
- INSERT
- SELECT
- UPDATE
- DELETE

# CREATE TABLE

- CAMPOS
  - Tipo
  - Valor por defecto
  - Restricciones (de columna): NOT NULL ...
- Restricciones (de tabla)
  - UNIQUE,
  - PRIMARY KEY, ...
  - CHECK ...
- HERENCIA (INHERITS...)
- CREATE TABLE AS SELECT ...

# CREATE INDEX

- (UNIQUE: valor único -no duplicados- del campo indexado)
- UNA O VARIAS COLUMNAS -o expresiones-
- METODOS
  - BTREE, RTREE, HASH, GIST
- EFICIENCIA
- (DROP INDEX)



# GRANT -REVOKE

- SELECT
- INSERT
- UPDATE
- DELETE
- REFERENCES
- RULE, TRIGGER ...
- ALL
- ON tabla TO usuario | GROUP grupo | PUBLIC

# INSERT

- inserción de una fila ... INSERT INTO tabla [(...)]
  - VALUES ( )
  - DEFAULT VALUES
- inserción de varias filas:
  - INSERT INTO tabla SELECT ....

# SELECT

- [ALL | DISTINCT [ ON expresión]]
- \* | expresión, ...
- FROM tabla alias, ... (SELECT ...) alias,...
  - las (tablas virtuales) requieren alias
- WHERE condición
- GROUP BY expresión, ...
- HAVING expresión de grupo ,...
- [ UNION | INTERSECT | EXCEPT ....]
- ORDER BY ...

# “JOINS”

- INNER JOIN (CROSS JOIN), OUTER JOIN

	PERFILES (Valores)					
	PIEZAS	IPE100	IPE200	IPE300	IPE400	PIEZAS (seccion)
●	SOP1-P1	x	x	x	x	SOP1-P1 IPE100
	SOP1-P2	x	x	x	x	SOP1-P2 IPE300
	SOP1-P3	x	x	x	x	SOP1-P3 IPE100
●	SOP2-P1	x	x	x	x	SOP2-P1 IPE300
	SOP2-P2	x	x	x	x	SOP2-P2 IPE400
	SOP3PP3	x	x	x	x	SOP3PP3
●	SOP4-P4	x	x	x	x	SOP4-P4 IPE400

- select pz.nombre, pf.nombre, pz.n/pf.area as tension from piezas pz, perfiles pf where pz.seccion=pf.nombre
- select ... from piezas pz LEFT OUTER JOIN perfiles pf ON ( pz.seccion=pf.nombre )

# CREATE VIEW

- CREATE VIEW AS SELECT ....
- VENTAJAS
  - ocultación -simplificación- de detalles
  - interfaz consistente
  - acceso público (GRANT) mejor acotado
- EFICIENCIA
  - búsqueda en cada momento
  - usar índices
- (DROP VIEW)

# CREATE RULE

- CREATE RULE nombre\_regla AS ON event TO tabla\_o\_vista [WHERE cond ] DO [ INSTEAD ] acción
  - event es SELECT, o INSERT, o UPDATE o DELETE
- permite añadir comprobaciones, insertar campos adicionales, como fecha o usuario, insertar referencias en vez de textos, hacer falsas inserciones en vistas ...

# UPDATE

- UPDATE tabla SET columna=expresion, ...  
WHERE ...
- UPDATE table SET columna=(SELECT ...) ...
  - UPDATE piezas SET tension = (SELECT  
piezas.normal/pf.area from perfiles pf where  
piezas.seccion=pf.nombre)
- UPDATE tabla SET columna=expresión, ...  
FROM tabla2 alias, ... WHERE ...
  - UPDATE piezas SET tension = piezas.normal/pf.area  
from perfiles pf where piezas.seccion=pf.nombre

# DELETE

- DELETE FROM tabla WHERE condición



# VACUUM

- VACUUM tabla
- VACUUM FULL ...
- VACUUM ANALYZE ...

# BEGIN

- BEGIN [WORK | TRANSACTION];
- ABORT
- ROLLBACK
- END
- COMMIT