



## CONSULTAS DE BASE DE DATOS EN POSTGRE SQL

EN ESTA SEGUNDA PARTE DEL PDF VAMOS A REALIZARLE UNAS CONSULTAS A NUESTRA BASE DE DATOS HOTEL QUE REALIZAMOS EN LA PRIMERA PARTE DEL PDF.

ACONTINUACION ENCONTRAREMOS LAS PREGUNTAS Y EL CODIGO PARA RESOLVER NUESTRAS CONSULTAS:

### I. INSERCIONES.

#### 1. INSERTE EL TIPO DE SERVICIO OCIO:

-- **OCIO:** SE LLAMA OCIO AL TIEMPO LIBRE QUE SE DEDICA A ACTIVIDADES QUE NO SON NI TRABAJO NI TAREAS DOMÉSTICAS ESENCIALES, Y PUEDEN SER RECREATIVAS. ES DIFERENTE AL TIEMPO DEDICADO A ACTIVIDADES OBLIGATORIAS COMO SON COMER, DORMIR.

```
INSERT INTO TIPO_SERVICIO (nombreservicio) VALUES ('OCIO');  
  
SELECT * FROM TIPO_SERVICIO;
```

✓ QUEDARA ASI:

	nombreservicio character varying(10)
1	COMEDOR
2	LAVANDERIA
3	OCIO

2. DE DE ALTA O REGISTRE UNA RESERVA DE LA HABITACIÓN 101 PARA EL CLIENTE 12345 PARA LAS NOCHES DEL 2 AL 4 DE JULIO DE 2009. EL CÓDIGO DE LA RESERVA ES AUTONUMÉRICO:

```
INSERT INTO RESERVA_HABITAC  
    (fechaentrada, fechasalida, iva, numhabitacion, cliente)  
VALUES  
    ('02-07-2009', '04-07-2009', 0.16, 101, '12345');
```



✓ QUEDARA ASI:

idreserva [PK] serial	fechaentrada date	fechasalida date	iva numeric(5,2)	numhabitacion integer	cliente character varying
1	2009-03-15	2009-03-25	0.07	101	12345
2	2009-03-15	2009-03-25	0.07	102	12345
3	2009-02-16	2009-02-21	0.07	103	12345
4	2009-03-16	2009-03-21	0.07	104	44444
5	2009-03-16	2009-03-21	0.07	105	44444
6	2009-03-16	2009-03-21	0.07	106	44444
7	2009-03-16	2009-03-21	0.07	107	44444
8	2009-07-02	2009-07-04	0.16	101	12345

## II. ACTUALIZACIONES.

1. ACTUALICE EL TELÉFONO DEL CLIENTE 12345. SU NUEVO NÚMERO ES 123456789:

```
SELECT * FROM CLIENTES WHERE identificacion = 12345;  
  
UPDATE CLIENTES SET telefono = '123456789' WHERE identificacion = 12345;
```

✓ QUEDARA ASI:

identificacion	pais	nombre	apellido1	apellido2	direccion	telefono	observaciones
character varying	character varying	character varying	character varying	character varying	character varying(30)	character varying	character varying
12345	ESPAÑA	Felipe	Iglesias	López	Avenida Los Castros, 44	123456789	Buen cliente

2. ACTUALICE EL PRECIO DE LOS SERVICIOS INCREMENTÁNDOLOS EN UN 2%:

```
UPDATE SERVICIOS SET precio = precio*1.02;
```

✓ QUEDARA ASI:



idreserv [PK]	nombreserv character vari	descripcion character varying(30)	precio money	iva numeric(5,2)	fecha date
1	COMEDOR	1 menu del Dia	€10,20	7.00	2009-01-01
2	LAVANDERIA	Lavado de Camisa	€2,04	7.00	2009-01-01
3	LAVANDERIA	Lavado de pantalon	€1,02	7.00	2009-01-01
4	LAVANDERIA	Lavado de pantalon	€861,90	7.00	2009-01-01

### III. BORRADOS.

1. BORRE LA RESERVA DE LA HABITACIÓN 101 REALIZADA ANTERIORMENTE:

```
DELETE FROM RESERVA_HABITAC WHERE idreserva = 8;
```

✓ QUEDARA ASI:

idreserva [PK] serial	fechaentrada date	fechasalida date	iva numeric(5,2)	numhabitac integer	cliente character
1	2009-03-15	2009-03-25	0.07	101	12345
2	2009-03-15	2009-03-25	0.07	102	12345
3	2009-02-16	2009-02-21	0.07	103	12345
4	2009-03-16	2009-03-21	0.07	104	44444
5	2009-03-16	2009-03-21	0.07	105	44444
6	2009-03-16	2009-03-21	0.07	106	44444
7	2009-03-16	2009-03-21	0.07	107	44444

➤ OTRA FORMA DE BORRAR:

```
DELETE FROM RESERVA_HABITAC  
WHERE cliente = 12345  
AND fechaentrada = '02-07-2009'  
AND numhabitacion = 101;
```

2. BORRE LOS TIPOS DE SERVICIO QUE NO TIENEN SERVICIOS DEFINIDOS:



```
DELETE FROM TIPO_SERVICIO
WHERE nombreservicio NOT IN (SELECT nombreservicio FROM SERVICIOS)
;
SELECT * FROM TIPO_SERVICIO;
```

✓ QUEDARA ASI:

nombreservicio character varying(10)
COMEDOR
LAVANDERIA

#### IV. CONSULTAS.

1. CREA UNA CONSULTA QUE DEVUELVA LOS CLIENTES CUYO APELLIDO1 INCLUYA LA SÍLABA "LE" ORDENADOS POR SU IDENTIFICADOR:

```
SELECT *
FROM CLIENTES
WHERE apellido1 LIKE '%le%'
ORDER BY identificacion;
```

✓ QUEDARA ASI:

identificacion	pais	nombre	apellido1	apellido2	direccion	telefono	observaciones
character va	character	character	character v	character	character varying(30)	character v	character vary
12345	ESPAÑA	Felipe	López	López	Avda Los Castros,	123456789	Buen cliente

2. CREA UNA CONSULTA QUE DEVUELVA LOS CLIENTES, ORDENADOS POR SU PRIMER APELLIDO, QUE TENGAN ALGUNA OBSERVACIÓN ANOTADA:

```
SELECT *
FROM CLIENTES
WHERE observaciones IS NOT NULL
ORDER BY apellido1
;
```

✓ QUEDARA ASI:



identificacion	pais	nombre	apellido1	apellido2	direccion	telefono	observaciones
character vary	character var	character vary	character va	character var	character varying(10)	character var	character vary
12345	ESPAÑA	Felipe	Iglesias	López	Avda Los Castros, 44	123456789	Buen cliente

3. CREA UNA CONSULTA QUE DEVUELVA LOS SERVICIOS CUYO PRECIO SUPERE LOS \$ 3  
ORDENADOS POR SU CÓDIGO DE SERVICIO:

```
SELECT * FROM servicios
WHERE (precio > '3')
ORDER BY idservicios
```

✓ QUEDARA ASI:

idservic	nombreservicio	descripcion	precio	iva	fecha
integer	character varying	character varying(30)	money	numeric(5,2)	date
1	COMEDOR	1 menu del Dia	€10,20	7.00	2009-01-01
4	LAVANDERIA	Lavado de pantalón	€861,90	7.00	2009-01-01

4. CREE UNA CONSULTA QUE DEVUELVA LAS HABITACIONES RESERVADAS PARA EL DÍA  
24 DE MARZO DE 2009:

--EL FORMATO DE LA FECHA DEBE SER IGUAL AL DE EL CAMPO EN LA BD.

**EJEM.** 'YYYY-MM-DD' ('2009-03-24') Y NO '24-032009'

```
SELECT numhabitacion
FROM RESERVA_HABITAC
WHERE '2009-03-24' BETWEEN fechaentrada AND fechasalida;
```

✓ QUEDARA ASI:

numhabita
integer
101
102



5. CREE UNA CONSULTA QUE DEVUELVA LOS CLIENTES PROCEDENTES DE ESPAÑA Y FRANCIA:

```
SELECT *  
FROM CLIENTES  
WHERE pais IN ('ESPAÑA', 'FRANCIA')  
;
```

✓ QUEDARA ASI:

identificacion	pais	nombre	apellido1	apellido2	direccion	telefono	observaciones
character vary	character vary	character vary	character vary	character vary	character varying(30)	character vary	character vary
12345	ESPAÑA	Felipe	Iglesias	López	Avda Los Castros,	123456789	Buen cliente
44444	ESPAÑA	Luis	García	García	Calle Mayor, 67	942456444	
456789	FRANCIA	Ludovic	Giuly	Bourquin	18 avenue Alsacen	37890194	

## V. CONSULTAS DE COMBINACIÓN.

1. CREA UNA CONSULTA QUE DEVUELVA LOS CLIENTES QUE HAN UTILIZADO EL SERVICIO DE COMEDOR:

```
SELECT SERVICIOS.NOMBRESERVICIO,  
SERVICIOS.IVA,  
SERVICIOS.FECHA,  
CLIENTES.NOMBRE,  
CLIENTES.APELLIDO1,  
GASTOS.IDSERVICIOS AS EXPR1  
FROM SERVICIOS INNER JOIN  
GASTOS ON SERVICIOS.idservicios = GASTOS.idservicios INNER JOIN  
RESERVA_HABITAC ON GASTOS.idreserva = RESERVA_HABITAC.idreserva INNER JOIN  
CLIENTES ON RESERVA_HABITAC.cliente = CLIENTES.identificacion  
WHERE (SERVICIOS.nombreservicio = 'COMEDOR')  
;
```

```
SELECT SER.nombreservicio,  
SER.iva,  
SER.fecha,  
CLI.nombre,  
CLI.apellido1,  
GAS.idservicios AS EXPR1  
FROM SERVICIOS SER,  
GASTOS GAS,  
RESERVA_HABITAC RH,  
CLIENTES CLI  
WHERE SER.nombreservicio = 'COMEDOR'  
AND SER.idservicios = GAS.idservicios  
AND GAS.idreserva = RH.idreserva  
AND RH.cliente = CLI.identificacion  
;
```

TRA FORMA:



✓ QUEDARA ASI:

nombreservicio	iva	fecha	nombre	apellido1	expr1
character varying(10)	numeric(5,2)	date	character varying(12)	character varying(12)	integer
COMEDOR	7.00	2009-	Felipe	Iglesias	1
COMEDOR	7.00	2009-	Felipe	Iglesias	1

2. CREA UNA CONSULTA QUE DEVUELVA LAS CARACTERÍSTICAS DE CADA HABITACIÓN RESERVADA:

```
SELECT TH.*,  
       HAB.numhabitacion  
FROM   HABITACIONES HAB,  
       TIPO_HABITACION TH,  
       RESERVA_HABITAC RH  
WHERE  HAB.numhabitacion = RH.numhabitacion  
       AND HAB.tipo_habitacion = TH.categoria  
;
```

✓ QUEDARA ASI:

categoria	camas	exterior	salon	terrazza	numhabit
integer	integer	character	charac	character	integer
1	1	SI	NO	NO	101
1	1	SI	NO	NO	102
1	1	SI	NO	NO	103
2	2	SI	NO	NO	104
2	2	SI	NO	NO	105
3	3	SI	NO	NO	106
4	1	SI	SI	NO	107





3. CREE UNA CONSULTA QUE DEVUELVA LOS PRECIOS DE LOS DISTINTOS TIPOS DE HABITACIÓN POR TEMPORADA:

```
SELECT PRECIO_HABITACION.tipo_habitacion,  
       PRECIO_HABITACION.precio,  
       TEMPORADA.fechainicio,  
       TEMPORADA.fechafinal,  
       TEMPORADA.tipo  
FROM   PRECIO_HABITACION INNER JOIN TEMPORADA ON PRECIO_HABITACION.temporada = TEMPORADA.temporada  
;
```

➤ OTRA FORMA:

```
SELECT PH.tipo_habitacion,  
       PH.precio,  
       TEMP.fechainicio,  
       TEMP.fechafinal,  
       TEMP.tipo  
FROM   PRECIO_HABITACION PH,  
       TEMPORADA TEMP  
WHERE  PH.temporada = TEMP.temporada  
;
```





✓ QUEDARA ASI:

tipo_habitacion integer	precio money	fechainicio date	fechafinal date	tipo charact
2	€40,00	2009-04-0	2009-05-	M
2	€45,00	2009-06-0	2009-08-	A
2	€40,00	2009-09-0	2009-10-	M
2	€35,00	2009-11-0	2009-12-	B
3	€40,00	2009-01-0	2009-03-	B
3	€45,00	2009-04-0	2009-05-	M
3	€50,00	2009-06-0	2009-08-	A
3	€45,00	2009-09-0	2009-10-	M
3	€40,00	2009-11-0	2009-12-	B
4	€50,00	2009-01-0	2009-03-	B
4	€55,00	2009-04-0	2009-05-	M
4	€60,00	2009-06-0	2009-08-	A
4	€55,00	2009-09-0	2009-10-	M
4	€50,00	2009-11-0	2009-12-	B
4	€523,0	2009-11-0	2009-12-	B

4. CREE UNA CONSULTA QUE DEVUELVA TODOS LOS CLIENTES, Y DE AQUELLOS QUE HAN REALIZADO ALGUNA RESERVA EN MARZO, INDICAR EL N° DE RESERVA:

--EL FORMATO DE LA FECHA DEBE SER IGUAL AL DE EL CAMPO EN LA BD.

EJEM. 'YYYY-MM-DD' ('2009-03-24') Y NO '24-032009'

```
SELECT CLI.*,
       RH.idreserva
FROM   CLIENTES CLI LEFT JOIN RESERVA_HABITAC RH ON CLI.identificacion = RH.cliente
       AND RH.fechaentrada BETWEEN '2009-03-01' AND '2009-03-30';
```

✓ QUEDARA ASI:

identificacion character vary	pais character	nombre character vary	apellido1 character v	apellido2 character	direccion character	telefono character va	observaciones character varyin	idreser integer
12345	ESPAÑA	Felipe	Iglesias	López	Avda Los	123456789	Buen cliente	2
12345	ESPAÑA	Felipe	Iglesias	López	Avda Los	123456789	Buen cliente	1
44444	ESPAÑA	Luis	García	García	Calle May	942456444		7
44444	ESPAÑA	Luis	García	García	Calle May	942456444		6
44444	ESPAÑA	Luis	García	García	Calle May	942456444		5
44444	ESPAÑA	Luis	García	García	Calle May	942456444		4
456789	FRANCIA	Ludovic	Giuly	Bourquin	18 avenue	37890194		



5. CREA UNA CONSULTA CON LOS SERVICIOS QUE NUNCA HAN SIDO CONTRATADOS:

```
SELECT DISTINCT *
FROM SERVICIOS
WHERE NOT EXISTS (SELECT * FROM GASTOS
                  WHERE GASTOS.IDSERVICIOS = SERVICIOS.IDSERVICIOS)
;
```

➤ OTRA FORMA:

```
SELECT DISTINCT SERVICIOS.*
FROM SERVICIOS
WHERE IDSERVICIOS NOT IN (SELECT IDSERVICIOS FROM GASTOS)
;
```

✓ QUEDARA ASI:

idservicios integer	nombreservicio character varying	descripcion character varying(30)	precio money	iva numeric(5,2)	fecha date
3	LAVANDERIA	Lavado de pantalon	€1,02	7.00	2009-01-01
4	LAVANDERIA	Lavado de pantalon	€861,9	7.00	2009-01-01

6. CREE UNA CONSULTA QUE DEVUELVA LOS CLIENTES CON EL MISMO PRIMER APELLIDO:

```
SELECT C.identificacion,
       C.apellido1,
       A.apellido1
FROM CLIENTES C,
     CLIENTES A
WHERE C.identificacion <> A.identificacion
      AND C.apellido1 = A.apellido1
;
```

➤ OTRA FORMA:

```
SELECT *
FROM CLIENTES;
INSERT INTO CLIENTES
VALUES ('22233', 'ESPAÑA', 'FELIPE', 'IGLESIAS', 'LÓPEZ', 'AVDA LOS CASTROS, 44', '942344444', 'BUEN CLIENTE');
DELETE FROM CLIENTES WHERE identificacion = '22233'
;
```



✓ QUEDARA ASI:

identificacion	apellido1	apellido1
character varying(12)	character varying(12)	character varying(12)

## VI. GROUP BY

1. CREA UNA CONSULTA QUE DEVUELVA EL N° DE CLIENTES POR NACIONALIDAD:

```
SELECT pais, COUNT(identificacion) AS NUMERO
FROM CLIENTES
GROUP BY pais
;
```

✓ QUEDARA ASI:

pais	numero
character varying(20)	bigint
FRANCIA	1
ESPAÑA	2

2. CREA UNA CONSULTA QUE DEVUELVA EL N° DE HABITACIONES POR CATEGORÍA DE HABITACIÓN:

```
SELECT COUNT(*),
       tipo_habitacion,
       camas,
       exterior
FROM HABITACIONES H,
     TIPO_HABITACION T
WHERE H.tipo_habitacion = T.categoria
GROUP BY tipo_habitacion, camas, exterior
;
```



✓ QUEDARA ASI:

count bigint	tipo_habitacion integer	camas integer	exterior character varying(2)
2	2	2	SI
1	4	1	SI
3	1	1	SI
1	3	3	SI

3. CREA UNA CONSULTA QUE DEVUELVA EL N° DE SERVICIOS QUE SE OFRECEN POR TIPO DE SERVICIO. RESTRINJA LA SALIDA PARA AQUELLOS TIPOS DE SERVICIO QUE OFREZCAN MÁS DE UN SERVICIO:

```
SELECT TIPO_SERVICIO.nombreservicio,  
       COUNT(idservicios) AS CANTIDAD  
FROM   TIPO_SERVICIO INNER JOIN SERVICIOS  
       ON TIPO_SERVICIO.nombreservicio = SERVICIOS.nombreservicio  
GROUP BY TIPO_SERVICIO.nombreservicio  
HAVING COUNT(*) <= 1  
;
```

➤ OTRAFORMA:

```
SELECT TIPO_SERVICIO.nombreservicio,  
       COUNT(idservicios) AS CANTIDAD  
FROM   SERVICIOS,  
       TIPO_SERVICIO  
WHERE  TIPO_SERVICIO.nombreservicio = SERVICIOS.nombreservicio  
GROUP BY TIPO_SERVICIO.nombreservicio  
HAVING COUNT(*) > 0  
;
```

✓ QUEDARA ASI:

nombreservicio character varying(10)	cantidad bigint
COMEDOR	1

4. CREE UNA CONSULTA QUE DEVUELVA EL GASTO EN SERVICIOS REALIZADO POR CADA RESERVA:



```
SELECT  GASTOS.idreserva,  
        SUM(GASTOS.cantidad * GASTOS.precio) AS TOTAL  
FROM    GASTOS  
GROUP BY GASTOS.idreserva  
;
```

✓ QUEDARA ASI:

idreserva integer	total money
4	€1.725
1	€20,00

## VII. OTRAS ACCIONES

1. CREE UNA CONSULTA QUE DEVUELVA EL PRECIO DEL SERVICIO MÁS CARO Y DEL MÁS BARATO:

```
SELECT  MAX(precio) AS MAXIMO,  
        MIN(precio) AS MÍNIMO  
FROM    SERVICIOS  
;
```

➤ OTRA FORMA:

```
SELECT  idservicios , 'MAXIMO', precio  
FROM    SERVICIOS  
WHERE   precio = (SELECT MAX(precio) FROM SERVICIOS)  
UNION  
SELECT  idservicios , 'MINIMO', precio  
FROM    SERVICIOS  
WHERE   precio = (SELECT MIN(precio) FROM SERVICIOS))  
;
```

✓ QUEDARA ASI:



idservicios integer	?column? text	precio money
3	MINIMO	€1,02
4	MAXIMO	€861,9

## VIII. VISTAS

CUANDO SE CREE UNA TABLA VISTA, ESTA DEBE COMENZAR CON EL PREFIJO  
IDX\_NOMBRETABLADESCRIPCION.

1. CREAR UNA VISTA QUE MUESTRE TODOS LOS DATOS DEL CLIENTE EXCEPTO LAS OBSERVACIONES:

```
CREATE VIEW IDX_CLTE_SINOBS AS  
SELECT  identificacion,  
        nombre,  
        apellido1,  
        apellido2,  
        direccion,  
        telefono  
FROM    CLIENTES
```

✓ QUEDARA ASI:

identificacion character vai	nombre character vai	apellido1 character vai	apellido2 character vai	direccion character vai	telefono character vai
12345	Felipe	Iglesias	López	Avda Los Ca	123456789
44444	Luis	García	García	Calle Mayor	942456444
456789	Ludovic	Giuly	Bourquin	18 avenue A	37890194





2. ACTUALIZAR EL NOMBRE DEL CLIENTE UTILIZANDO ESTA VISTA:

```
UPDATE CLIENTES SET nombre = 'CAMBIO' WHERE identificacion = 456789;  
UPDATE CLIENTES SET nombre = 'LUDOVIC' WHERE identificacion = 456789;
```

✓ QUEDARA ASI:

```
LINE 1: ...CLIENTES SET nombre = 'CAMBIO' WHERE identificacion = 456789;  
.  
HINT: Ningún operador coincide con el nombre y el tipo de los argumentos. Puede ser necesario agregar conversiones explícitas de tipos.  
.  
***** Error *****  
.  
ERROR: el operador no existe: character varying = integer  
SQL state: 42083  
Hint: Ningún operador coincide con el nombre y el tipo de los argumentos. Puede ser necesario agregar conversiones explícitas de tipos.  
Character: 60
```

3. CREAR UNA VISTA QUE MUESTRE EL ID, NOMBRE Y PRIMER APELLIDO DE TODOS LOS CLIENTES. ACTUALIZAR EL NOMBRE DEL CLIENTE UTILIZANDO ESTA VISTA. ¿SE PUEDE?:

```
CREATE VIEW IDX_CLTE_DATOS AS  
SELECT CLI.identificacion,  
       CLI.nombre,  
       CLI.apellido1  
FROM   CLIENTES CLI LEFT JOIN RESERVA_HABITAC RH  
       ON CLI.identificacion = RH.cliente  
GROUP BY identificacion, nombre, apellido1  
;  
UPDATE IDX_CLTE_DATOS SET NOMBRE = 'CAMBIO' WHERE IDENTIFICACION = 456789  
;
```

✓ QUEDARA ASI: NO SE PUEDE ACTUALIZAR UNA VISTAS.

```
ERROR: el operador no existe: character varying = integer  
LINE 9: ...TE_DATOS SET NOMBRE = 'CAMBIO' WHERE IDENTIFICACION = 456789  
.  
HINT: Ningún operador coincide con el nombre y el tipo de los argumentos. Puede ser necesario agregar conversiones explícitas de tipos.  
.  
***** Error *****  
.  
ERROR: el operador no existe: character varying = integer  
SQL state: 42083  
Hint: Ningún operador coincide con el nombre y el tipo de los argumentos. Puede ser necesario agregar conversiones explícitas de tipos.  
Character: 267
```

IX. CHECK





1. CONTROLE EN LA TABLA TEMPORADA QUE LA FECHA DE FIN ES MAYOR QUE LA FECHA DE INICIO:

```
ALTER TABLE TEMPORADA ADD CONSTRAINT TEM_FECHA CHECK (fechainicio < fechafinal);
```

✓ QUEDARA ASI:

```
-- Check: tem_fecha  
  
-- ALTER TABLE temporada DROP CONSTRAINT tem_fecha;  
  
ALTER TABLE temporada  
ADD CONSTRAINT tem_fecha CHECK (fechainicio < fechafinal);
```

2. CONTROLE EN LA TABLA RESERVA\_HABITAC QUE LA FECHA DE SALIDA ES MAYOR O IGUAL QUE LA FECHA DE ENTRADA:

```
ALTER TABLE RESERVA_HABITAC ADD CONSTRAINT RH_FECHA CHECK (fechaentrada < fechasalida);
```

✓ QUEDARA ASI:

```
-- Check: rh_fecha  
  
-- ALTER TABLE reserva_habitac DROP CONSTRAINT rh_fecha;  
  
ALTER TABLE reserva_habitac  
ADD CONSTRAINT rh_fecha CHECK (fechaentrada < fechasalida);
```

3. CONTROLE EN LA TABLA SERVICIOS QUE EL IVA ESTÁ COMPRENDIDO ENTRE 0 Y 100,0:

```
ALTER TABLE SERVICIOS ADD CONSTRAINT SERV_IVA CHECK (iva >= 0 AND iva <= 100);
```

✓ QUEDARA ASI:

```
-- Check: serv_iva  
  
-- ALTER TABLE servicios DROP CONSTRAINT serv_iva;  
  
ALTER TABLE servicios  
ADD CONSTRAINT serv_iva CHECK (iva >= 0::numeric AND iva <= 100::numeric);
```

## X. CREATE INDEX

1. CREE UN ÍNDICE SOBRE EL ATRIBUTO NACIONALIDAD DE LA TABLA CLIENTES, CON OBJETO DE AGILIZAR LAS BÚSQUEDAS:



```
CREATE INDEX IDX_CLTE_PAIS ON CLIENTES(pais);
```

✓ QUEDARA ASI:

```
-- Index: idx_clte_pais  
-- DROP INDEX idx_clte_pais;  
  
CREATE INDEX idx_clte_pais  
ON clientes  
USING btree  
(pais COLLATE pg_catalog."default" );
```

2. CREE UN ÍNDICE SOBRE EL ATRIBUTO NUMHABITACION DE LA TABLA REVERVA\_HABITAC, CON OBJETO DE AGILIZAR LAS BÚSQUEDAS:

```
CREATE INDEX IDX_HAB_NUM ON HABITACIONES(numhabitacion);
```

✓ QUEDARA ASI:

```
-- Index: idx_hab_num  
-- DROP INDEX idx_hab_num;  
  
CREATE INDEX idx_hab_num  
ON habitaciones  
USING btree  
(numhabitacion );
```

3. CREE UN ÍNDICE SOBRE EL ATRIBUTO NOMBRESERVICIO DE LA TABLA SERVICIOS, CON OBJETO DE AGILIZAR LAS BÚSQUEDAS.

```
CREATE INDEX IDX_SERV_NOM ON SERVICIOS(nombreservicio);
```

✓ QUEDARA ASI:



```
-- Index: idx_serv_nom  
-- DROP INDEX idx_serv_nom;  
  
CREATE INDEX idx_serv_nom  
ON servicios  
USING btree  
(nombreservicio COLLATE pg_catalog."default" );
```

## XII. ALTER TABLE

1. AÑADA UNA COLUMNA CALCULADA EN LA TABLA GASTOS QUE OBTENGA EL IMPORTE POR LÍNEA SIN APLICAR EL IVA.

```
ALTER TABLE GASTOS ADD COLUMN importe MONEY AS (cantidad * precio)
```

✓ QUEDARA ASI: **NO FUNCIONA**

```
ERROR: error de sintaxis en o cerca de «AS»  
LINE 1: ALTER TABLE GASTOS ADD COLUMN importe MONEY AS (cantidad * p...
```

```
***** Error *****
```

```
ERROR: error de sintaxis en o cerca de «AS»  
SQL state: 42601  
Character: 45
```

2. AÑADA UNA COLUMNA EN SERVICIOS QUE INDIQUE SI EL SERVICIO SE OFRECE A LOS CLIENTES O NO. EL CAMPO BIT ES DE LONG 1:

```
ALTER TABLE SERVICIOS ADD COLUMN ofertado BIT NOT NULL DEFAULT '1';
```

✓ QUEDARA ASI:

idservicios [PK] integer	nombreservi character vai	descripcion character vai	precio money	iva numeric(5,2)	fecha date	ofertado bit(1)
1	COMEDOR	1 menu del	€10,20	7.00	2009-01-01	1
2	LAVANDERIA	Lavado de C	€2,04	7.00	2009-01-01	1
3	LAVANDERIA	Lavado de p	€1,02	7.00	2009-01-01	1
4	LAVANDERIA	Lavado de p	€861,90	7.00	2009-01-01	1



## FIN

- OTRO EJEMPLO DE VISTA, PARA LA BD FACTURA:



```
CREATE VIEW V_CATEPROD
AS SELECT CAT.catg_nom AS CATEGORIA, PRO.prod_des AS PRODUCTO
FROM CATG_PRODUCTO CAT, PRODUCTO PRO
WHERE CAT.id_catg = PRO.id_catg
GROUP BY CAT.catg_nom, PRO.prod_des
;
SELECT * from v_cateprod
;
CREATE VIEW V_CATEPROD2
AS SELECT CAT.catg_nom AS CATEGORIA, SUM(PRO.prod_val_uni) AS PRODUCTO
FROM CATG_PRODUCTO CAT, PRODUCTO PRO
WHERE CAT.id_catg = PRO.id_catg
GROUP BY CAT.catg_nom
;
SELECT * from v_cateprod2
```

## AHORA SI FIN