

```
1  create database gym
2  go
3
4  use gym
5  go
6
7  /* Tablas Maestras */
8
9  /* Personas */
10
11 create table Personas(
12 Persona_ID int primary key identity (0, 1),
13 Nombre nvarchar(50) not null,
14 Apellido nvarchar(50) not null,
15 Tipo_Documento_ID int not null,
16 Nro_documento varchar(50) not null,
17 Tipo_Sexo_ID int not null,
18 Nro_Telefono nvarchar(24) not null,
19 Nro_Alternativo nvarchar(24) null,
20 Mail nvarchar(50) null,
21 Observaciones nvarchar(200) null,
22 Fecha_Alta date not null,
23 Fecha_Baja date null,
24 )
25
26 go
27
28 create table Empleados(
29 Empleado_ID int primary key identity (0, 1),
30 Persona_ID int not null,
31 Usuario nvarchar(255) null,
32 Clave varchar(200) null,
33 Tipo_Empleado_ID int not null,
34 Estado_Empleado_ID int not null
35 )
36
37 go
38
39 /*
40 0 Activo
41 1 Vacaciones
42
43 */
44
45 create table Clientes(
46 Cliente_ID int primary key identity (0, 1),
47 Persona_ID int not null,
48 Estado nvarchar(1) not null
49 )
50
51 go
52
53 create table Asistencias(
54 Asistencia_ID int primary key identity (0, 1),
55 Fecha datetime not null,
56 Estado nvarchar(1) not null, --P(resente), T(arde), A(usente)
57 Cliente_ID int not null,
58 Empleado_ID int not null, --quien hizo el registro
59 Plan_Asignado_ID int not null
60 )
61
```

```
62 go
63
64 /*
65     Falta tabla donde se guarde cada persona durante cada fecha de clase
66     Falta una tabla que maneje fechas
67     1 Tabla: (por día de clase)
68     alumno, fechaInscripción, anulaPlan, fechaAnulacion (del día del turno) y asistencia
69     2 Tabla: Generar distintos horarios para cada fecha.
70     Jornadas, Planes
71 */
72 -- fecha de alta generación de plan, falta en PlanAsignado
73 --
74
75 /* Caja */
76
77 create table Cajas(
78 Caja_ID int primary key identity (0, 1),
79 Empleado_ID_Apertura int not null,
80 Fecha_Apertura date not null,
81 Hora_Apertura time null,
82 Importe_Apertura decimal(18,0) not null,
83 Caja_Abierta bit not null,
84 Empleado_ID_Cierre int null,
85 Fecha_Cierre date null,
86 Hora_Cierre time null,
87 Importe_Cierre decimal(18,0) null,
88 Importe_Cierre_Caja decimal (18, 0) null
89 )
90
91 go
92
93 create table Detalles_Cajas(
94 Detalle_Caja_ID int primary key identity (0, 1),
95 Caja_ID int not null,
96 Plan_Asignado_ID int null,
97 Empleado_ID int not null,
98 Importe_Ingreso decimal (18, 0) null,
99 Importe_Egreso decimal (18, 0) null,
100 Observaciones nvarchar(200) null,
101 )
102
103 go
104
105 ----Se complica realizar un inner join de las cuotas,
106 ----debido a los ID. Por lo que se va a modificar y este
107 ----ID de persona, va a corresponder al cliente.
108 --create table Cuotas(
109 --Cuota_ID int primary key identity (0, 1),
110 --Plan_Asignado_ID int not null,
111 --Empleado_ID int not null,
112 --Cliente_ID int not null,
113 --Detalle_Caja_ID int null,
114 --Importe_Cuota decimal(18, 0) not null,
115 --Saldo decimal(18, 0) not null,
116 --Vto_Cuota date not null
117 --)
118
119 go
120
121 /* Planes */
122
```

```
123 create table Planes(  
124 Plan_ID int primary key identity (0, 1),  
125 Persona_ID int not null, /*quien hace el registro*/  
126 Empleado_ID int null, /* el profesor */  
127 Nombre nvarchar(20) not null,  
128 Importe_Plan decimal (18, 0) not null,  
129 Duracion int null,  
130 Cupo_Total int null,  
131 Cupo_Restante int null,  
132 Estado nvarchar (1) not null,  
133 Fecha_Alta_Plan date null,  
134 Fecha_Inicio date null, /* fecha de inicio */  
135 Fecha_Fin date null /* fecha de cierre */  
136 )  
137  
138 go  
139  
140 create table Planes_Asignados(  
141 Plan_Asignado_ID int primary key identity (0, 1),  
142 Plan_ID int not null,  
143 Cliente_ID int not null,  
144 Empleado_ID int not null,  
145 Fecha_Inscripcion date not null,  
146 Estado nvarchar(1) not null  
147 )  
148  
149 go  
150  
151 -- ¿Cómo voy a saber si tengo saldo en Plan asignado, si no tengo  
152 -- una tabla que me diga que tiene un saldo vencido?  
153 -- Para eso creo una tabla que sea de la facturación.  
154 -- Se genera de manera automática a principio de mes  
155 -- Y el programa va a saber así si hay una cuota vencida o pendiente de pago.  
156 --create table Facturas_Clientes(  
157 --Factura_Cliente_ID int primary key identity (0, 1),  
158 --Plan_Asignado_ID int not null,  
159 --Empleado_ID int not null,  
160 --Importe decimal(18, 0) not null,  
161 --Fecha_Emision date not null,  
162 --Fecha_Vencimiento date not null,  
163 --Saldo decimal (18, 0) not null --Acá se le agregar a la facturación anterior.  
164 --)  
165  
166 --go  
167  
168 /* Jornadas */  
169  
170 create table Jornadas_Planes(  
171 Jornada_Plan_ID int primary key identity (0, 1),  
172 Plan_ID int not null,  
173 Dia nvarchar(10) not null,  
174 Desde_Hora datetime not null,  
175 Hasta_Hora datetime not null,  
176 Estado nvarchar(1) not null  
177 )  
178  
179 go  
180  
181 create table Jornadas_Empleados(  
182 Jornada_Empleado_ID int primary key identity (0, 1),  
183 Empleado_ID int not null,
```

```
184 Dia nvarchar(10) not null,
185 Desde_Hora datetime not null,
186 Hasta_Hora datetime not null,
187 Estado nvarchar(1) not null
188 )
189
190 go
191
192
193 /* Registro de Login */
194
195 create table Registros_Logs(
196 Registro_Log_ID int primary key identity (0, 1),
197 Empleado_ID int not null,
198 Fecha_LogIn datetime not null,
199 Fecha_LogOut datetime null
200 )
201
202 go
203
204 /* Fin Tablas Maestras */
205
206 /* Tablas Tipo */
207
208 create table Tipos_Documentos(
209 Tipo_Documento_ID int primary key identity (0, 1),
210 Tipo varchar(4) not null,
211 Estado varchar(8) not null
212 )
213
214 go
215
216 insert into Tipos_Documentos values ('DNI', 'Activo')
217 insert into Tipos_Documentos values ('LC', 'Activo')
218 insert into Tipos_Documentos values ('LE', 'Activo')
219 insert into Tipos_Documentos values ('CI', 'Activo')
220 insert into Tipos_Documentos values ('PAS', 'Activo')
221 insert into Tipos_Documentos values ('EXT', 'Activo')
222 insert into Tipos_Documentos values ('CUIL', 'Activo')
223 insert into Tipos_Documentos values ('CUIT', 'Activo')
224
225 go
226
227 create table Tipos_Sexos(
228 Tipo_Sexo_ID int primary key identity (0, 1),
229 Sexo varchar(20) not null,
230 Estado varchar(8) not null
231 )
232
233 go
234
235 insert into Tipos_Sexos values ('Masculino', 'Activo')
236 insert into Tipos_Sexos values ('Femenino', 'Activo')
237 insert into Tipos_Sexos values ('Otros', 'Activo')
238
239 go
240
241 create table Tipos_Empleados(
242 Tipo_Empleado_ID int primary key identity (0, 1),
243 Tipo varchar(20) not null,
244 Acceso_Clave varchar(1) not null,
```

```
245 Estado varchar(8) not null
246 )
247
248 go
249
250 insert into Tipos_Empleados values ('Admin', 'Y', 'Activo')
251 insert into Tipos_Empleados values ('Jefe', 'Y', 'Activo')
252 insert into Tipos_Empleados values ('Usuario', 'Y', 'Activo')
253 insert into Tipos_Empleados values ('Profesor', 'N', 'Activo')
254 insert into Tipos_Empleados values ('Otros', 'N', 'Activo')
255
256 go
257
258 create table Estados_Empleados(
259 Estado_Empleado_ID int primary key identity (0, 1),
260 Estado_Empleado varchar(20) not null,
261 Estado varchar(8) not null
262 )
263
264 go
265
266 insert into Estados_Empleados values ('Activo', 'Activo')
267 insert into Estados_Empleados values ('Vacaciones', 'Activo')
268 insert into Estados_Empleados values ('Licencia Médica', 'Activo')
269 insert into Estados_Empleados values ('Baja', 'Activo')
270
271 go
272
273 /* Fin Tablas Tipo */
274
275 /* Constrains */
276
277 /* Personas */
278
279
280 alter table Personas
281 add CONSTRAINT FK_Personas_TipoDocumento FOREIGN KEY (Tipo_Documento_ID)
282 REFERENCES Tipos_Documentos (Tipo_Documento_ID)
283
284 go
285
286 alter table Personas
287 add CONSTRAINT FK_Personas_TipoSexo FOREIGN KEY (Tipo_Sexo_ID)
288 REFERENCES Tipos_Sexos (Tipo_Sexo_ID)
289
290 go
291
292 /* Empleados */
293
294 alter table Empleados
295 add CONSTRAINT FK_Empleado_Tipo_Empleado FOREIGN KEY (Tipo_Empleado_ID)
296 REFERENCES Tipos_Empleados (Tipo_Empleado_ID)
297
298 go
299
300 alter table Empleados
301 add CONSTRAINT FK_Empleado_Estado_Empleado FOREIGN KEY (Estado_Empleado_ID)
302 REFERENCES Estados_Empleados (Estado_Empleado_ID)
303
304 go
305
```

```
306 alter table Empleados
307 add CONSTRAINT FK_Empleado_Persona FOREIGN KEY (Persona_ID)
308 REFERENCES Personas (Persona_ID)
309
310 go
311
312
313 /* Clientes */
314
315
316 alter table Clientes
317 add CONSTRAINT FK_Cliente_Persona FOREIGN KEY (Persona_ID)
318 REFERENCES Personas (Persona_ID)
319
320 go
321
322 /* Asistencias */
323
324 alter table Asistencias
325 add CONSTRAINT FK_Asistencia_Cliente FOREIGN KEY (Cliente_ID)
326 REFERENCES Clientes (Cliente_ID)
327
328 go
329
330 alter table Asistencias
331 add CONSTRAINT FK_Asistencia_Empleado FOREIGN KEY (Empleado_ID)
332 REFERENCES Empleados (Empleado_ID)
333
334 go
335
336 /* Caja */
337
338 alter table Cajas
339 add CONSTRAINT FK_Cajas_Empleado_Apertura FOREIGN KEY (Empleado_ID_Apertura)
340 REFERENCES Empleados (Empleado_ID)
341
342 go
343
344 alter table Cajas
345 add CONSTRAINT FK_Cajas_Empleado_Cierre FOREIGN KEY (Empleado_ID_Cierre)
346 REFERENCES Empleados (Empleado_ID)
347
348 go
349
350 /* Detalles de Cajas */
351
352 alter table Detalles_Cajas
353 add CONSTRAINT FK_Detalles_Cajas FOREIGN KEY (Caja_ID)
354 REFERENCES Cajas (Caja_ID)
355
356 go
357
358 alter table Detalles_Cajas
359 add CONSTRAINT FK_Detalles_Cajas_Empleado FOREIGN KEY (Empleado_ID)
360 REFERENCES Empleados (Empleado_ID)
361
362 go
363
364 /* Cuotas */
365
366 --alter table Cuotas
```

```
367 --add CONSTRAINT FK_Cuotas_Plan_Asignado FOREIGN KEY (Plan_Asignado_ID)
368 --REFERENCES Planes_Asignados (Plan_Asignado_ID)
369
370 --go
371
372 --alter table Cuotas
373 --add CONSTRAINT FK_Cuotas_Clientes FOREIGN KEY (Cliente_ID)
374 --REFERENCES Clientes (Cliente_ID)
375
376 --go
377
378 --alter table Cuotas
379 --add CONSTRAINT FK_Cuotas_Empleados FOREIGN KEY (Empleado_ID)
380 --REFERENCES Empleados (Empleado_ID)
381
382 --go
383
384 --alter table Cuotas
385 --add CONSTRAINT FK_Nombre_Detalles_Cajas FOREIGN KEY (Detalle_Caja_ID)
386 --REFERENCES Detalles_Cajas (Detalle_Caja_ID)
387
388 --go
389
390 /* Planes */
391
392 alter table Planes
393 add CONSTRAINT FK_Planes_Persona_Logueada FOREIGN KEY (Persona_ID)
394 REFERENCES Personas (Persona_ID)
395
396 go
397
398 alter table Planes
399 add CONSTRAINT FK_Planes_Profesor_De_La_Clase FOREIGN KEY (Empleado_ID)
400 REFERENCES Empleados (Empleado_ID)
401
402 go
403
404 /* Planes_Asignados */
405
406 alter table Planes_Asignados
407 add CONSTRAINT FK_Planes_Asignados_Planes FOREIGN KEY (Plan_ID)
408 REFERENCES Planes (Plan_ID)
409
410 go
411
412 alter table Planes_Asignados
413 add CONSTRAINT FK_Planes_Asignados_Cliente FOREIGN KEY (Cliente_ID)
414 REFERENCES Clientes (Cliente_ID)
415
416 go
417
418 alter table Planes_Asignados
419 add CONSTRAINT FK_Planes_Asignados_Empleados FOREIGN KEY (Empleado_ID)
420 REFERENCES Empleados (Empleado_ID)
421
422 go
423
424 /* Facturas_Clientes */
425
426 --alter table Facturas_Clientes
427 --add CONSTRAINT FK_Facturas_Clientes_Constrain FOREIGN KEY (Plan_Asignado_ID)
```

```
428 --REFERENCES Planes_Asignados (Plan_Asignado_ID)
429
430 --go
431
432 --alter table Facturas_Clientes
433 --add CONSTRAINT FK_Facturas_Clientes_Cliente FOREIGN KEY (Empleado_ID)
434 --REFERENCES Empleados (Empleado_ID)
435
436 --go
437
438 /* Jornadas_Planes */
439
440 alter table Jornadas_Planes
441 add CONSTRAINT FK_Jornada_Planes FOREIGN KEY (Plan_ID)
442 REFERENCES Planes (Plan_ID)
443
444 go
445
446 /* Jornada_Empleados */
447
448 alter table Jornadas_Empleados
449 add CONSTRAINT FK_Jornada_Empleados FOREIGN KEY (Empleado_ID)
450 REFERENCES Empleados (Empleado_ID)
451
452 go
453
454 /* Registro de Login */
455
456 alter table Registros_Logs
457 add CONSTRAINT FK_Registros_Logs_Empleado FOREIGN KEY (Empleado_ID)
458 REFERENCES Empleados (Empleado_ID)
459
460 go
461
462 /* Fin Constrains */
463
464
465 /* Procedimientos, triggers y dem s */
466
467 create proc sp_cargar_tipos_documentos
468 as
469 select Tipo_Documento_ID, Tipo
470 from Tipos_Documentos
471 where Estado = 'Activo'
472
473 go
474
475 create proc sp_cargar_tipos_sexos
476 as
477 select Tipo_sexo_ID, Sexo
478 from Tipos_Sexos
479 where Estado = 'Activo'
480
481 go
482
483 create proc sp_cargar_tipos_empleados
484 as
485 select Tipo, Tipo_Empleado_ID
486 from Tipos_Empleados
487 where Tipo != 'Admin'
488 and Tipo != 'Jefe'
```



```
489
490 go
491
492 create proc sp_cargar_tipos_empleados_Jefe
493 as
494 select Tipo_Empleado_ID, Tipo
495 from Tipos_Empleados
496 where Tipo != 'Admin'
497 and Estado = 'Activo'
498
499 go
500
501 create proc sp_cargar_tipos_empleados_Admin
502 as
503 select Tipo_Empleado_ID, Tipo
504 from Tipos_Empleados
505 where Estado = 'Activo'
506
507 go
508
509
510 create proc sp_cargar_Profesores_Actuales
511 as
512 select p.Nombre, e.Empleado_ID
513 from Personas as p
514 inner join Empleados as e
515 on e.Persona_ID = p.Persona_ID
516 inner join Tipos_Empleados as tp
517 on e.Tipo_Empleado_ID = tp.Tipo_Empleado_ID
518 where tp.Tipo = 'Profesor'
519 and e.Estado_Empleado_ID = 0
520
521 go
522
523 create proc sp_cargar_ultimo_ID
524 as
525 select Empleado_ID
526 from Empleados
527 where Empleado_ID = (select(max(Empleado_ID)) from Empleados)
528
529 go
530
531 create proc sp_cargar_ultimo_ID_Personas
532 as
533 select Persona_ID
534 from Personas
535 where Persona_ID = (select(max(Persona_ID)) from Personas)
536
537 go
538
539 create proc sp_Cargar_Clientes_Desc
540 as
541 SELECT c.Cliente_ID,
542 p.Persona_ID, p.Nombre, p.Apellido, p.Nro_Documento, p.Nro_telefono,
543 c.Estado
544 FROM Personas as p
545 INNER JOIN Clientes as c
546 ON p.Persona_ID = c.Persona_ID
547 WHERE p.Fecha_Alta BETWEEN GETDATE()-7 AND GETDATE()
548 and c.Estado = 'A'
549 ORDER BY p.Fecha_Alta desc
```

```
550
551 go
552
553 create proc sp_Cargar_Empleados_Desc
554 AS
555 SELECT p.Persona_ID, p.Nombre, p.Apellido, p.Nro_Documento,
556        tp.Tipo, ep.Estado_Empleado
557 FROM Personas as p
558 INNER JOIN Empleados as em
559 ON p.Persona_ID = em.Persona_ID
560 INNER JOIN Tipos_Empleados tp
561 ON em.Tipo_Empleado_ID = tp.Tipo_Empleado_ID
562 INNER JOIN Estados_Empleados as ep
563 ON em.Estado_Empleado_ID = ep.Estado_Empleado_ID
564 WHERE p.Fecha_Alta BETWEEN GETDATE()-7 AND GETDATE()
565 and em.Tipo_Empleado_ID != 0
566 and em.Tipo_Empleado_ID != 1
567 ORDER BY p.Fecha_Alta desc
568
569 go
570
571 create procedure sp_last_Login
572 as
573 select Registro_Log_ID
574 from Registros_Logs
575 where Registro_Log_ID = (select(max(Registro_Log_ID)) from Registros_Logs)
576
577 go
578
579 create proc sp_cargar_planes
580 as
581 select Plan_ID, Nombre
582 from Planes
583 where Estado = 'A'
584
585 go
586
587 create procedure sp_cargar_planes_Actuales
588 as
589 select pl.Plan_ID, pl.Nombre, p.Nombre, replace(pl.Estado, 'A', 'Activo') as Estado
590 from Planes as pl
591 inner join Empleados as e
592 on e.Empleado_ID = pl.Empleado_ID
593 inner join Personas as p
594 on p.Persona_ID = e.Persona_ID
595 inner join Tipos_Empleados tp
596 on e.Tipo_Empleado_ID = tp.Tipo_Empleado_ID
597 where pl.Estado = 'A'
598 and tp.Tipo = 'Profesor'
599 or pl.Estado = (select replace(pl.Estado, 'I', 'Inactivo') as Estado)
600
601 go
602
603 create procedure sp_get_Asistencias_diarias
604 as
605 select c.Cliente_ID, pe.Nombre, pe.Apellido, pe.Nro_documento, pl.Nombre, a.Estado
606 from Asistencias as a
607 inner join Clientes as c
608 on c.Cliente_ID = a.Cliente_ID
609 inner join Personas as pe
610 on c.Persona_ID = pe.Persona_ID
```

```
611 inner join Planes_Asignados as pa
612 on a.Plan_Asignado_ID = pa.Plan_Asignado_ID
613 inner join Planes as pl
614 on pl.Plan_ID = pa.Plan_ID
615 where a.Fecha between GETDATE() - 1 and GETDATE()
616 order by a.Fecha desc
617
618 go
619
620 create procedure sp_Buscar_Listado_Asistencia_Total @Parametro nvarchar
621 as
622 begin
623 select distinct(c.Cliente_ID) as Cliente_ID, p.Nombre, p.Apellido, p.Nro_documento,
pl.Nombre, a.Estado
624 from Asistencias as a
625 inner join Clientes as c
626 on c.Cliente_ID = a.Cliente_ID
627 inner join Personas as p
628 on c.Persona_ID = p.Persona_ID
629 inner join Planes_Asignados pa
630 on a.Plan_Asignado_ID = pa.Plan_Asignado_ID
631 inner join Planes pl
632 on pl.Plan_ID = pa.Plan_ID
633 where p.Nombre like @Parametro
634 or p.Apellido like @Parametro
635 or a.Fecha like @Parametro
636 or pl.Nombre like @Parametro
637 end
638
639 go
640
641 create procedure sp_get_cajas
642 as
643 select c.Caja_ID, c.Fecha_Apertura, c.Importe_Apertura, c.Importe_Cierre,
644 dc.Importe_Ingreso, dc.Importe_Egreso, dc.Motivo,
645 p.Apellido
646 from Cajas as c
647 left join Detalles_Cajas as dc
648 on dc.Caja_ID = c.Caja_ID
649 and dc.Empleado_ID = c.Empleado_ID_Apertura
650 left join Empleados as em
651 on em.Empleado_ID = c.Empleado_ID_Apertura
652 inner join Personas as p
653 on p.Persona_ID = em.Persona_ID
654 order by c.Fecha_Apertura desc
655
656 go
657
658
659 create procedure sp_Cargar_Presentes @Plan_ID int, @Fecha date
660 as
661 begin
662 set dateformat dmy;
663 Select p.Nombre + ' ' + p.Apellido as NombreCliente, c.Cliente_ID as Cliente_ID
664 from Personas as p
665 left join Clientes as c
666 on p.Persona_ID = c.Persona_ID
667 inner join Planes_Asignados as pa
668 on c.Cliente_ID = pa.Cliente_ID
669 inner join Planes as pl
670 on pa.Plan_ID = pl.Plan_ID
```

```
671 left join Asistencias as a
672 on a.Cliente_ID = c.Cliente_ID
673 where pl.Plan_ID = @Plan_ID
674 and a.Fecha = @Fecha
675 and a.Cliente_ID is not null
676 end
677
678 go
679
680 create procedure sp_Cargar_Ausentes @Plan_ID int, @Fecha date
681 as
682 begin
683 set dateformat dmy;
684 Select p.Nombre + ' ' + p.Apellido as NombreCliente, c.Cliente_ID
685 from Personas as p
686 inner join Clientes as c
687 on p.Persona_ID = c.Persona_ID
688 inner join Planes_Asignados as pa
689 on c.Cliente_ID = pa.Cliente_ID
690 inner join Planes as pl
691 on pa.Plan_ID = pl.Plan_ID
692 where pl.Plan_ID = @Plan_ID
693 and c.Cliente_ID not in (select Cliente_ID from Asistencias where Fecha = @Fecha)
694 end
695
696 go
697
698 create procedure sp_Buscar_Cliente_Ausentismo @buscar nvarchar
699 as
700 begin
701 select pa.Plan_Asignado_ID, c.Cliente_ID,
702 p.Nombre, p.Apellido, p.Nro_documento, p.Nro_Telefono, p.Mail,
703 pl.Nombre
704 from Personas as p
705 inner join Clientes c
706 on c.Persona_ID = p.Persona_ID
707 inner join Planes_Asignados as pa
708 on c.Cliente_ID = pa.Cliente_ID
709 inner join Planes as pl
710 on pa.Plan_ID = pl.Plan_ID
711 where pa.Estado = 'A'
712 and c.Estado = 'A'
713 and p.Nro_documento like @buscar
714 end
715
716 go
717
718
719 create procedure sp_Buscar_Cliente_Por_ID @Plan_ID int, @Cliente_ID int
720 as
721 begin
722 select pa.Plan_Asignado_ID, c.Cliente_ID,
723 p.Nombre, p.Apellido, p.Nro_documento, p.Nro_Telefono, p.Mail,
724 pl.Nombre
725 from Personas as p
726 inner join Clientes as c
727 on c.Persona_ID = p.Persona_ID
728 inner join Planes_Asignados as pa
729 on c.Cliente_ID = pa.Cliente_ID
730 inner join Planes pl
731 on pa.Plan_ID = pl.Plan_ID
```

```
732 where pa.Estado = 'A'
733 and c.Estado = 'A'
734 and pa.Plan_ID = @Plan_ID
735 and c.Cliente_ID = @Cliente_ID
736 end
737
738 go
739
740 create procedure sp_Buscar_Listado_Clientes @query nvarchar
741 as
742 begin
743 SELECT c.Cliente_ID, p.Persona_ID, p.Nombre,
744 p.Apellido, p.Nro_Documento, p.Nro_telefono,
745 c.Estado
746 FROM Personas as p
747 INNER JOIN Clientes as c
748 ON p.Persona_ID = c.Persona_ID
749 where p.Nombre LIKE @query
750 or p.Apellido LIKE @query
751 or p.Nro_Documento LIKE @query
752 or c.Estado LIKE @query
753 and c.Estado = 'A'
754 Order by p.Fecha_Alta desc
755 end
756
757 go
758
759 create procedure sp_Calcular_Importes @Caja_ID int
760 as
761 begin
762 select c.Importe_Apertura, sum(dc.Importe_Ingreso) as Ingreso, sum(dc.Importe_Egreso) as
Egreso
763 from Cajas as c
764 inner join Detalles_Cajas as dc
765 on dc.Caja_ID = c.Caja_ID
766 where c.Caja_ID = @Caja_ID
767 group by c.Importe_Apertura, dc.Importe_Ingreso, dc.Importe_Egreso
768 end
769
770
771 go
772
773 create procedure sp_Get_Cajas_Y_Detalles @Parametro nvarchar
774 as
775 begin
776 select Caja_ID, Fecha_Apertura, Importe_Apertura,
777 Importe_Cierre_Caja, Importe_Cierre, sum(Importe_Cierre_Caja - Importe_Cierre) as
Diferencia
778 from Cajas
779 where Caja_ID like @Parametro
780 or Fecha_Apertura like @Parametro
781 group by Caja_ID, Fecha_Apertura, Importe_Apertura, Importe_Cierre_Caja,
Importe_Cierre
782 order by Fecha_Apertura desc
783 end
784
785
786 go
787
788 create procedure sp_cargar_Plan_unico @Estado nvarchar, @Profesor nvarchar, @Parametro
nvarchar
789 as begin
790 select pl.Plan_ID, pl.Nombre as Nombre_Planes, p.Nombre as Nombre_Empleado,
```

```
replace(pl.Estado, 'A', 'Activo') as Estado
791 from Planes as pl
792 inner join Empleados em
793 on em.Empleado_ID = pl.Empleado_ID
794 inner join Personas as p
795 on p.Persona_ID = em.Persona_ID
796 inner join Tipos_Empleados as tp
797 on em.Tipo_Empleado_ID = tp.Tipo_Empleado_ID
798 where pl.Estado = @Estado
799 and tp.Tipo = @Profesor
800 or pl.Estado = (select replace(pl.Estado, 'I', 'Inactivo') as Estado)
801 and pl.Nombre Like @Parametro
802 or p.Nombre like @Parametro
803 or pl.Estado like @Parametro
804 end
805
806 go
807
808 create procedure sp_Get_Last_Caja_ID @Fecha date
809 as begin
810 select max(Caja_ID) as Caja_ID from Cajas
811 where Caja_Abierta = 1
812 and Fecha_Apertura = @Fecha
813 end
814
815 go
816
817
818 create procedure sp_Planes_Cliente_Para_Pago @Documento nvarchar, @Estado nvarchar
819 as begin
820 select pl.Plan_ID, pl.Nombre
821 from Planes as pl
822 inner join Personas as pe
823 on pl.Persona_ID = pe.Persona_ID
824 where pe.Nro_documento like @Documento
825 and pl.Estado = @Estado
826 end
827
828 go
829
830 create procedure sp_Buscar_Planes_Asignados_Cliente @Estado nvarchar, @Cliente_ID int
831 as begin
832 Select pl.Nombre
833 from Planes_Asignados as pa
834 inner join Planes as pl
835 on pl.Plan_ID = pa.Plan_ID
836 where pl.Estado = @Estado
837 and pa.Cliente_ID = @Cliente_ID
838 end
839
840 go
841
842 create procedure sp_Detalle_Caja_Diario
843 as begin
844 select dc.Detalle_Caja_ID, dc.Importe_Ingreso, dc.Importe_Egreso, dc.Observaciones
845 From Detalles_Cajas as dc
846 inner join Cajas as ca
847 on ca.Caja_ID = dc.Caja_ID
848 where ca.Caja_Abierta = 1
849 end
850
```

```
851 go
852
853 create procedure sp_Detalle_Caja_Diario_Con_Parametro @Parametro nvarchar
854 as begin
855 select dc.Detalle_Caja_ID, dc.Importe_Ingreso, dc.Importe_Egreso, dc.Observaciones
856 From Detalles_Cajas as dc
857 inner join Cajas as ca
858 on ca.Caja_ID = dc.Caja_ID
859 where ca.Caja_Abierta = 1
860 and ca.Fecha_Apertura like @Parametro
861 or dc.Importe_Ingreso like @Parametro
862 or dc.Importe_Egreso like @Parametro
863 or dc.Observaciones like @Parametro
864 end
865
866 go
867
868 /* queries para realizar registros */
869
870 create procedure sp_abrir_caja @Empleado_ID_Apertura int, @Fecha datetime,
871 @Importe_Apertura decimal, @Caja_Abierta bit
872 as
873 insert into Cajas (Empleado_ID_Apertura, Fecha_Apertura, Hora_Apertura, Importe_Apertura,
874 Caja_Abierta)
875 values(@Empleado_ID_Apertura, @Fecha, @Fecha, @Importe_Apertura, @Caja_Abierta)
876
877 go
878
879 create procedure sp_Cerrar_Caja @Empleado_ID_Cierre int, @Fecha_Cierre DateTime,
880 @Importe_Cierre decimal, @Importe_Cierre_Caja decimal, @Caja_ID int, @Caja_Abierta bit
881 as
882 begin
883 update Cajas set
884 Empleado_ID_Cierre = @Empleado_ID_Cierre,
885 Fecha_Cierre = @Fecha_Cierre,
886 Hora_Cierre = @Fecha_Cierre,
887 Importe_Cierre = @Importe_Cierre,
888 Importe_Cierre_Caja = @Importe_Cierre_Caja,
889 Caja_Abierta = @Caja_Abierta
890 where Caja_ID = @Caja_ID
891 end
892
893 /* - - */
```