set search\_path = unc\_esq\_peliculas;  
  
--Listar todas las películas que poseen entregas de películas de idioma inglés durante  
--el año 2006. (P)  
select \*  
from pelicula p  
where p.codigo\_pelicula in (select r.codigo\_pelicula  
 from renglon\_entrega r  
 where r.nro\_entrega in (select e.nro\_entrega  
 from entrega e  
 where *extract*(year from fecha\_entrega) = 2006  
 )  
 )AND p.idioma ilike 'Inglés';  
  
SELECT \*  
FROM pelicula p  
WHERE p.codigo\_pelicula IN (  
 SELECT codigo\_pelicula FROM renglon\_entrega r  
 WHERE r.nro\_entrega IN (  
 SELECT nro\_entrega FROM entrega e  
 WHERE *EXTRACT*(YEAR FROM fecha\_entrega) = 2006  
 )  
) AND idioma ILIKE 'Inglés';  
  
-- 1.2) Indicar la cantidad de películas que han sido entregadas en 2006 por un distribuidor  
-- nacional. Trate de resolverlo utilizando ensambles.(P)  
SELECT distinct p.codigo\_pelicula, *count*(\*),p.titulo  
FROM pelicula p  
JOIN renglon\_entrega r on p.codigo\_pelicula = r.codigo\_pelicula  
JOIN entrega e on r.nro\_entrega = e.nro\_entrega  
JOIN distribuidor d on e.id\_distribuidor = d.id\_distribuidor  
WHERE d.tipo = 'N' AND *EXTRACT*(YEAR FROM e.fecha\_entrega) >= 2006  
GROUP BY p.codigo\_pelicula;  
  
--Indicar los departamentos que no posean empleados cuya diferencia de sueldo  
--máximo y mínimo (asociado a la tarea que realiza) no supere el 40% del sueldo máximo.  
--(P) (Probar con 10% para que retorne valores)  
SELECT \*  
FROM departamento d  
WHERE (id\_departamento,id\_distribuidor) NOT IN  
 (SELECT DISTINCT id\_departamento, id\_distribuidor  
 FROM empleado p  
 WHERE p.id\_tarea IN  
 (SELECT t.id\_tarea  
 FROM tarea t  
 WHERE (t.sueldo\_maximo - t.sueldo\_minimo) <= (t.sueldo\_maximo \* 0.1)));  
  
set search\_path = unc\_esq\_peliculas;  
  
--1.4. Liste las películas que nunca han sido entregadas por un distribuidor nacional.(P)  
SELECT DISTINCT \*  
FROM pelicula p  
WHERE p.codigo\_pelicula IN(  
 SELECT r.codigo\_pelicula  
 FROM renglon\_entrega r  
 WHERE r.nro\_entrega IN(  
 SELECT e.nro\_entrega  
 FROM entrega e  
 WHERE e.id\_distribuidor IN(  
 SELECT d.id\_distribuidor  
 FROM distribuidor d  
 WHERE d.id\_distribuidor NOT IN(  
 SELECT n.id\_distribuidor  
 FROM nacional n  
 WHERE d.tipo ='N'  
 )  
 )  
 )  
 )  
--1.5. Determinar los jefes que poseen personal a cargo y cuyos departamentos (los del  
--jefe) se encuentren en la Argentina.  
SELECT DISTINCT e.nombre,e.apellido,e.id\_jefe,e.id\_empleado  
FROM empleado e  
JOIN departamento d ON e.id\_departamento = d.id\_departamento  
JOIN ciudad c ON d.id\_ciudad = c.id\_ciudad  
WHERE e.id\_jefe IS NULL AND c.id\_pais ILIKE '%AR%';  
  
/\*1.6. Liste el apellido y nombre de los empleados que pertenecen a aquellos  
departamentos de Argentina y donde el jefe de departamento posee una comisión de más  
del 10% de la que posee su empleado a cargo.\*/  
SELECT DISTINCT e.apellido,e.nombre,e.id\_jefe,d.nombre  
FROM empleado e  
JOIN departamento d ON e.id\_departamento = d.id\_departamento AND  
 e.id\_distribuidor = d.id\_distribuidor  
JOIN empleado jefe ON d.jefe\_departamento = jefe.id\_empleado  
JOIN ciudad c ON d.id\_ciudad = c.id\_ciudad  
WHERE jefe.porc\_comision > (e.porc\_comision \* 1.1) AND c.id\_pais ILIKE '%AR%';  
  
SELECT e.nombre,e.apellido  
FROM empleado e  
WHERE (id\_departamento,id\_distribuidor) IN  
 (SELECT d.id\_departamento,d.id\_distribuidor  
 FROM departamento d JOIN empleado jefe ON d.jefe\_departamento = jefe.id\_empleado  
 WHERE d.id\_ciudad IN(  
 SELECT c.id\_ciudad  
 FROM ciudad c  
 WHERE jefe.porc\_comision > (e.porc\_comision \* 1.1) AND c.id\_pais ILIKE 'AR'  
 )  
 );  
  
/\*1.7. Indicar la cantidad de películas entregadas a partir del 2010, por género. \*/  
  
SELECT *count*(DISTINCT p.codigo\_pelicula)  
FROM pelicula p  
JOIN renglon\_entrega r ON p.codigo\_pelicula=r.codigo\_pelicula  
JOIN entrega e ON r.nro\_entrega=e.nro\_entrega  
WHERE *extract*(year FROM e.fecha\_entrega) >= 2010  
GROUP BY p.genero  
  
/\* Realizar un resumen de entregas por día, indicando el video club al cual se le  
realizó la entrega y la cantidad entregada. Ordenar el resultado por fecha.\*/  
SELECT e.fecha\_entrega,v.id\_video,*SUM*(r.cantidad)  
FROM entrega e  
JOIN renglon\_entrega r ON e.nro\_entrega = r.nro\_entrega  
JOIN video v ON e.id\_video = v.id\_video  
GROUP BY e.fecha\_entrega,v.id\_video  
ORDER BY e.fecha\_entrega  
  
/\*1.9. Listar, para cada ciudad, el nombre de la ciudad y la cantidad de empleados  
mayores de edad que desempeñan tareas en departamentos de la misma y que posean al  
menos 30 empleados. \*/  
SELECT c.nombre\_ciudad,c.id\_pais, *count*(e.id\_empleado) as cantidad  
FROM ciudad c  
JOIN departamento d ON c.id\_ciudad = d.id\_ciudad  
JOIN empleado e ON d.id\_distribuidor = e.id\_distribuidor AND d.id\_departamento = e.id\_departamento  
JOIN tarea t ON e.id\_tarea = t.id\_tarea  
WHERE *extract*(year FROM fecha\_nacimiento) <= 2005  
group by c.nombre\_ciudad,c.id\_pais  
HAVING *count*(e.id\_empleado) >=30  
  
SELECT c.nombre\_ciudad  
FROM ciudad c  
WHERE c.id\_ciudad IN  
 (SELECT d.id\_ciudad  
 FROM departamento d  
 WHERE (d.id\_distribuidor,d.id\_departamento) IN  
 (SELECT e.id\_distribuidor,e.id\_departamento  
 FROM empleado e  
 WHERE e.id\_tarea IN  
 (SELECT t.id\_tarea  
 FROM tarea t  
 WHERE *extract*(year FROM fecha\_nacimiento) <= 2005  
 group by c.nombre\_ciudad  
 HAVING *count*(e.id\_empleado) >=30  
 )  
 )  
 );  
  
/\*Muestre, para cada institución, su nombre y la cantidad de voluntarios que realizan  
aportes. Ordene el resultado por nombre de institución. V \*/  
SELECT i.nombre\_institucion, *count*(v.nro\_voluntario)  
FROM institucion i  
JOIN voluntario v ON i.id\_institucion=v.id\_institucion  
WHERE v.horas\_aportadas IS NOT NULL  
GROUP BY i.nombre\_institucion  
ORDER BY i.nombre\_institucion

/\*2.2. Determine la cantidad de coordinadores en cada país, agrupados por nombre de  
país y nombre de continente. Etiquete la primer columna como &#39;Número de coordinadores&#39;  
\*/  
SELECT *count*(v.id\_coordinador) as Numero\_Coordinadores, p.nombre\_pais,c.nombre\_continente  
FROM voluntario v  
JOIN institucion i ON v.id\_institucion=i.id\_institucion  
JOIN direccion d ON i.id\_direccion=d.id\_direccion  
JOIN pais p ON d.id\_pais = p.id\_pais  
JOIN continente c ON p.id\_continente=c.id\_continente  
GROUP BY p.nombre\_pais,c.nombre\_continente  
  
/\*2.3. Escriba una consulta para mostrar el apellido, nombre y fecha de nacimiento de  
cualquier voluntario que trabaje en la misma institución que el Sr. de apellido Zlotkey.  
Excluya del resultado a Zlotkey. \*/  
set search\_path = unc\_esq\_voluntario;  
  
SELECT v.nombre, v.apellido, v.fecha\_nacimiento  
FROM voluntario v inner join institucion i on v.id\_institucion = i.id\_institucion  
where v.apellido not like 'Zlotkey' AND v.id\_institucion in (  
 SELECT v2.id\_institucion  
 FROM voluntario v2  
 where v2.apellido ilike 'Zlotkey'  
 )  
  
/\*Cree una consulta para mostrar los números de voluntarios y los apellidos de todos  
los voluntarios cuya cantidad de horas aportadas sea mayor que la media de las horas  
aportadas. Ordene los resultados por horas aportadas en orden ascendente. \*/  
SELECT v.nro\_voluntario,v.apellido  
FROM voluntario v  
WHERE v.horas\_aportadas > (SELECT *avg*(v2.horas\_aportadas)  
 FROM voluntario v2)  
ORDER BY v.horas\_aportadas

set search\_path = unc\_252169;  
CREATE TABLE distribuidor\_nac  
(  
id\_distribuidor numeric(5,0) NOT NULL,  
nombre character varying(80) NOT NULL,  
direccion character varying(120) NOT NULL,  
telefono character varying(20),  
nro\_inscripcion numeric(8,0) NOT NULL,  
encargado character varying(60) NOT NULL,  
id\_distrib\_mayorista numeric(5,0),  
CONSTRAINT pk\_distribuidorNac PRIMARY KEY (id\_distribuidor)  
);  
  
set search\_path = unc\_esq\_voluntario  
select DISTINCT d.nombre,d.direccion,d.telefono,n.nro\_inscripcion,n.encargado,n.id\_distrib\_mayorista  
from distribuidor d  
JOIN nacional n ON d.id\_distribuidor = n.id\_distribuidor  
  
/\*3.1 Se solicita llenarla con la información correspondiente a los datos completos de  
todos los distribuidores nacionales. \*/  
INSERT INTO unc\_252169.distribuidor\_nac (id\_distribuidor, nombre, direccion, telefono, nro\_inscripcion, encargado, id\_distrib\_mayorista)  
SELECT d.id\_distribuidor, d.nombre, d.direccion, d.telefono, n.nro\_inscripcion, n.encargado, n.id\_distrib\_mayorista  
FROM unc\_esq\_peliculas.distribuidor d  
JOIN unc\_esq\_peliculas.nacional n ON d.id\_distribuidor = n.id\_distribuidor;  
  
/\*Agregar a la definición de la tabla distribuidor\_nac, el campo "CODIGO PAIS" que  
indica el código de país del distribuidor mayorista que atiende a cada distribuidor  
nacional.(codigo\_pais varchar(5) NULL) \*/  
ALTER TABLE distribuidor\_nac  
ADD codigo\_pais varchar(5) NULL;  
  
INSERT INTO distribuidor\_nac (codigo\_pais)  
SELECT i.codigo\_pais  
FROM unc\_esq\_peliculas.internacional i  
JOIN unc\_esq\_peliculas.nacional d on i.id\_distribuidor = d.id\_distribuidor  
/\*3.3. Para todos los registros de la tabla distribuidor\_nac, llenar el nuevo campo  
&quot;codigo\_pais&quot; con el valor correspondiente existente en la tabla &quot;Internacional&quot;.  
3.4. Eliminar de la tabla distribuidor\_nac los registros que no tienen asociado un  
distribuidor mayorista. \*/  
  
delete from distribuidor\_nac n  
where n.id\_distrib\_mayorista IS null;

\*Cómo debería implementar las Restricciones de Integridad Referencial  
 (RIR) si se desea que cada vez que se elimine un registro de la tabla  
 PALABRA , también se eliminen los artículos  
 que la referencian en la tabla CONTIENE. \*/  
ALTER TABLE p5p1e1\_contiene  
DROP CONSTRAINT fk\_p5p1e1\_contiene\_palabra; --elimino a restric anterior  
  
ALTER TABLE P5P1E1\_CONTIENE  
ADD CONSTRAINT FK\_P5P1E1\_CONTIENE\_PALABRA  
FOREIGN KEY (idioma, cod\_palabra)  
REFERENCES P5P1E1\_PALABRA (idioma, cod\_palabra)  
ON DELETE CASCADE;  
  
delete from p5p1e1\_palabra where cod\_palabra=1;  
  
insert into p5p1e1\_palabra  
(idioma, cod\_palabra, descripcion)  
VALUES ('ES', 1, 'hola'),  
 ('IN', 2, 'adios'),  
 ('ES', 3, 'hola22');  
  
insert into p5p1e1\_contiene  
(id\_articulo, idioma, cod\_palabra)  
values (1,'IN',2),  
 (1,'ES',1);  
  
insert into p5p1e1\_articulo  
(id\_articulo, titulo, autor)  
values (1,'la fuga','picazzo');  
  
/\*b) Verifique qué sucede con las palabras contenidas en cada artículo, al eliminar una palabra,  
si definen la Acción Referencial para las bajas (ON DELETE) de la RIR correspondiente  
como:  
ii) Restrict  
iii) Es posible para éste ejemplo colocar SET NULL o SET DEFAULT para ON  
DELETE y ON UPDATE?  
 1\_restric esta bien  
 2\_se puede colocar null si acepta nulos en este caso no y default si ya hay un valor por default\*/  
alter table p5p1e1\_contiene  
add constraint fk\_contiene\_palabra  
foreign key (cod\_palabra,idioma)  
references p5p1e1\_palabra (cod\_palabra,idioma)  
on delete restrict ;  
  
  
-- Table: TP5\_P1\_EJ2\_AUSPICIO  
ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_EMPLEADO  
 FOREIGN KEY (tipo\_empleado, nro\_empleado)  
 REFERENCES TP5\_P1\_EJ2\_EMPLEADO (tipo\_empleado, nro\_empleado)  
 MATCH FULL  
 ON DELETE SET NULL  
 ON UPDATE RESTRICT  
;  
-- Reference: FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO (table: TP5\_P1\_EJ2\_AUSPICIO)  
ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO  
 FOREIGN KEY (id\_proyecto)  
 REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)  
 ON DELETE RESTRICT  
 ON UPDATE RESTRICT  
;  
-- Reference: FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_EMPLEADO (table: TP5\_P1\_EJ2\_TRABAJA\_EN)  
ALTER TABLE TP5\_P1\_EJ2\_TRABAJA\_EN ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_EMPLEADO  
 FOREIGN KEY (tipo\_empleado, nro\_empleado)  
 REFERENCES TP5\_P1\_EJ2\_EMPLEADO (tipo\_empleado, nro\_empleado)  
 ON DELETE CASCADE  
 ON UPDATE RESTRICT  
;  
-- Reference: FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_PROYECTO (table: TP5\_P1\_EJ2\_TRABAJA\_EN)  
ALTER TABLE TP5\_P1\_EJ2\_TRABAJA\_EN ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_PROYECTO  
 FOREIGN KEY (id\_proyecto)  
 REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)  
 ON DELETE RESTRICT  
 ON UPDATE CASCADE  
;