

1.

Dadas las relaciones R y S calcular:

- $R \cup S$
- $R - S$
- $R \times S$
- $R \bowtie_x S$
- $\pi_B(R)$
- $\sigma_{A=C}(R \times S)$
- $S \div T$
- $R \bowtie_x S$
 $R.B < S.C$

R: A B	S: B C	T: C
-----	-----	--
a b	b c	c
b c	e a	d
c b	b d	
d e		

2.

Dada la siguiente Base de Datos de una Facultad de Ciencias:

- **ALUMNO** (NroLU, Nombre, Edad, Sexo, FeclIngreso)
 - **MATERIA** (IdMateria, Nombre)
 - **CURSA** (NroLU, IdMateria)
 - **CARRERA** (IdCarrera, Nombre)
 - **DOCENTE** (Legajo, Nombre)
 - **DICTA** (Legajo, IdMateria)
 - **PLAN** (IdPlan, Nombre, IdCarrera)
 - **PLAN_DE_ESTUDIO** (IdPlan, IdMateria)
- Mediante SQL, AR y CRT listar los nombres y edades de los alumnos mujeres que ingresaron en el año 2009.
 - Mediante SQL, AR y CRT listar para cada alumno las materias que cursa. Se deberá mostrar el nombre del alumno y el nombre de la materia.
 - Mediante SQL, AR y CRT listar para cada alumno el nombre de la carrera que cursa.
 - Mediante AR y CRT obtener los nombres de las materias que tienen más de 1 alumno hombre mayor de 30 años.
 - Mediante SQL obtener los nombres de las materias que tienen más de 50 alumnos hombres mayores de 30 años. ¿Es posible resolver esta consulta usando AR y CRT? Justifique la respuesta.
 - Mediante SQL indicar cuántas materias tiene cada Plan de estudios de la carrera Ciencias de la Computación.
 - Mediante SQL y AR listar los nombres de los alumnos mayores de 25 años que cursan todas las materias de la carrera 'Ciencias de la Computación' que no son dictadas por el profesor 'Jeff Ullman'.
 - Mediante SQL indicar para cada materia en cuántos planes de estudio se requiere cursar dicha materia.
 - Mediante SQL resolver el punto b) incluyendo en el listado los alumnos que no están cursando ninguna materia.

- j) Mediante SQL listar los nombres de las materias que tienen más de 30 alumnos inscriptos.

NOTA: Los nombres y las descripciones no se repiten. No hay valores nulos. Sexo puede ser 'M' o 'F'. Una materia puede ser dictada por más de un docente.

3.

Dada la siguiente Base de Datos:

- **ELEMENTO** (nro_e, desc, costo)
- **PARTE_DE** (nro_e, nro_e_1), donde nro_e_1 es un componente de nro_e.
- **PROVEE** (nro_p, nro_e)
- **PROVEEDOR** (nro_p, nombre, domicilio)

Escribir en **AR** y en **CRT** la siguiente consulta:

Listar los códigos de los elementos que componen todos los elementos provistos por proveedores que no proveen elementos de costo superior a \$ 100.

NOTA: Considerar que, por ej., si e1 es componente de e2, y e2 es componente de e3, entonces e1 es componente de e3. (esta es la máxima profundidad admitida).

4.

Considerando el siguiente esquema de una base de datos:

- **FRECUENTA**(Persona, Bar)
- **SIRVE**(Bar, Cerveza)
- **GUSTA**(Persona, Cerveza).

Expresar las siguientes consultas usando álgebra relacional (**AR**):

- Bares que sirven alguna cerveza que le guste a "Juan K."
- Personas que frecuentan al menos un bar que sirve alguna cerveza que les guste.
- Personas que no frecuenten ningún bar que sirva una cerveza que les guste.
- Personas que frecuentan todos los bares. (Asumir que todos los bares sirven al menos una cerveza).

5.

Considerando el siguiente esquema de una base de datos:

- **CLIENTES**(Nombre Cliente, Dirección, Saldo)
- **ORDENES**(Numero, Nombre_Cliente, Ítem, Cantidad): contiene los pedidos de los clientes.
- **PROVEEDORES**(Nombre Proveedor, Ítem, Precio)

Expresar las siguientes consultas y operaciones utilizando **SQL**:

- Listado de todos los clientes indicando las cantidades totales pedidas por ellos.
- Nombres de los proveedores sobre cuyos ítems se tengan órdenes por más de 2000 unidades en total.
- Nombres de los clientes que ordenaron más de 15 lámparas.

- d. Nombres de los clientes cuya cantidad total de unidades pedidas se encuentren por encima del promedio general de unidades pedidas de todos los clientes.
- e. Actualizar en un 10% la cantidad de ítems pedidos en las ordenes de aquellos clientes cuyo saldo sea mayor a 10.000.

6.

Sea el siguiente esquema de relación:

- **PERSONA** (NOMBRE, PROGENITOR),
 - **AMIGO**(NOMBRE1, NOMBRE2).
El atributo PROGENITOR indica para cada PERSONA el NOMBRE de su padre y puede tener valor NULL.
- a. Suponga que la relación AMIGO es tal que $(a,a) \notin \text{AMIGO}$ para ningún a y que si $(a,b) \in \text{AMIGO}$ entonces que $(b,a) \notin \text{AMIGO}$.
 - I. Expresar en el **álgebra relacional** y en **cálculo relacional de tuplas** la consulta que devuelve como resultado a aquellas personas cuyos padres no tienen amigos.
 - II. Expresar en el **álgebra relacional** y en **cálculo relacional de tuplas** la consulta que devuelve como resultado a aquellas personas cuyos amigos son amigos de sus padres.
 - III. Expresar una consulta **SQL** que devuelva como resultado a las personas que tengan mas de 8 amigos
 - b. Suponga que se ha descuidado el control de las tuplas que se ingresan a la base de datos, y que la restricción de antisimetría expresada en a) ya no es válida. Escriba la consulta en SQL que en esta situación resuelve lo pedido en el punto a.3.

7.

Considerando el siguiente esquema de una base de datos:

- **MIEMBRO** (Código_M, Apellido, Nombre, Especialidad)
 - **EVENTO** (Código_E, Especialidad, Nombre_E, Fecha, Fecha_limite_inscripción, Hora_curso, Duración, Cupo_mínimo, Código_A)
 - **AUDITORIO** (Código_A, Nombre_A, Dirección, Localidad, Capacidad, Fecha_Disponible_Desde, Fecha_Disponible_Hasta)
 - **PARTICIPA** (Código_M, Código_E)
 - **ORGANIZA** (Código_M, Código_E)
- * Los miembros participan, organizan eventos o ambas cosas.

Expresar las siguientes consultas utilizando SQL:

- a. Obtener todos los datos de los organizadores que no participan de ningún evento.
- b. Recuperar la cantidad de eventos por localidad que sean posibles de realizar y que traten de temas médicos (tener en cuenta cupo mínimo y cantidad de inscriptos).
- c. Obtener los datos de los cursos que deben suspenderse por falta de quórum. Asumir la existencia de una función built-in TODAY() que devuelve la fecha del sistema.
- d. Obtener los nombres de los miembros que participan de todos los eventos.
- e. Borrar los auditorios que no hayan sido utilizados en ningún evento.

8.

Dados los siguientes esquemas relacionales:

- **PERSONA**(Nombre, Sexo, Fecha_Nac, Nombre_Madre, Nombre_Padre),
- **MATRIMONIO**(Nombre_Mujer, Nombre_Hombre, Fecha_Realización, Fecha_Fin)

Se asume que dos personas pueden contraer matrimonio entre sí más de una vez.

- a. Listar en SQL las personas que hayan tenido algún hijo en su primer matrimonio.
- b. Expresar en **cálculo relacional de tuplas** la consulta que devuelve como resultado los nombres de los hijos extramatrimoniales.
- c. Expresar en **álgebra relacional** la consulta que devuelve como resultado los nombres de los primogénitos de cada matrimonio. Recordar que en álgebra relacional no se dispone de funciones de agregación.

9.

Dados los siguientes esquemas relacionales:

- **PERSONA**(Nombre, Sexo, Fecha_Nac),
 - **ENFERMEDAD**(Nombre_Paciente, Enfermedad, Fecha_Inicio, Fecha_Fin)
- a. Listar en **SQL** la enfermedad más común para los chicos que tienen entre 5 y 10 años de edad. Suponer que el operador - permite la sustracción de fechas, y que existe una función del sistema TODAY.
 - b. Listar en **SQL** la enfermedad más común para cada edad.
 - c. Expresar en **AR** los nombres de los pacientes que han contraído más de una enfermedad simultáneamente.

10.

Sean los siguientes esquemas de relación:

- **ARTICULO** (cod_arti, nom_arti)
- **VENDEDOR** (cod_vende, nom_vende)
- **PRECIO** (cod_arti, cod_vende, f_vigencia, valor)

Asuma que todo artículo que vende un vendedor tiene un precio, y que un vendedor nunca deja de proveer un artículo una vez que se hace cargo de su venta.

Resolver en **álgebra relacional** (cuando sea posible) y en **SQL** las siguientes consultas:

- a. Obtener los nombres de los vendedores que venden todos los artículos al día 29/01/03.
- b. Obtener el menor precio del artículo de nombre X ese mismo día.
- c. Actualizar en un 5% el valor de aquellos artículos que son vendidos por el vendedor Carlos.

11.

Una cadena de comercios cuenta con un depósito centralizado donde se almacenan los productos comprados por las distintas sucursales (c/u de ellas los adquiere en forma independiente) y luego los van retirando en la medida de sus necesidades. La actividad del depósito se registra en el siguiente esquema relacional:

- **SUCURSAL** (cod_suc, nom_suc, zona)
- **PRODUCTO** (cod_prod, nom_prod, stock_critico)
Obs: *stock_critico* es la cantidad mínima deseada para TODO el depósito (NO para cada sucursal).
- **MOVIMIENTO** (nro_mov, fecha_hora, cod_suc, signo_mov)
Obs: *fecha_hora* es diferente para cada movimiento.
signo_mov es +1 si es un depósito ó -1 si es un retiro.
- **MOV_PROD** (nro_mov, cod_prod, cantidad)

Considerando que el esquema satisface las siguientes condiciones de integridad referencial:

- **MOVIMIENTO**(cod_suc) referencia **SUCURSAL**(cod_suc)
 - **MOV_PROD**(nro_mov) referencia **MOVIMIENTO**(nro_mov)
 - **MOV_PROD**(cod_prod) referencia **PRODUCTO**(cod_prod)
- a. En AR y SQL, obtener el código y el nombre de las sucursales que el mes de abril del 2010 hayan enviado a depósito TODOS los productos que comercializa la cadena (se encuentran en PRODUCTO) y que también hayan realizado retiros de TODOS los productos.
Obs: Los productos no necesariamente se han depositado en un único movimiento sino en varios. Lo mismo vale para los retiros.
 - b. En SQL, obtener los códigos y los nombres de los productos que algún retiro del mes de abril del 2010 haya dejado con un total almacenado en el depósito por debajo de su stock crítico.
Obs: El stock de un producto en un momento dado es igual a la suma (con signo) de todos los movimientos de ese producto realizados hasta dicho momento.
 - c. En AR (sin funciones de agregación) y SQL (sin vistas ni sub-queries en el FROM) obtener el código y el nombre de los productos tales que, durante el mes de abril del 2010: toda sucursal haya retirado en al menos dos oportunidades y no haya realizado ningún depósito (las dos condiciones deben ser satisfechas por cada sucursal).
 - d. En SQL (sin vistas ni sub-queries en el FROM), obtener el número de movimiento, el código y nombre de la sucursal asociada de aquellos depósitos del mes de abril del 2010 tales que: la cantidad depositada de cada uno de los ítems que lo componen es superior al stock del producto correspondiente para la sucursal asociada.
Por ejemplo: si la sucursal S en el movimiento X depositó los productos A, B y C; entonces la cantidad depositada de A es superior al stock hasta ese momento del producto A para la sucursal S (dicho stock es: la suma con signo de todos los movimientos del producto A para S), ídem para B e ídem para C.

12.

Dados los siguientes esquemas relacionales:

- **Producto**(codigo_prod, nombre_prod)
- **Vendedor**(legajo, nombre_vend)
- **Cliente**(cuit, nombre_clie)
- **Asignacion**(cuit_clie, legajo_vend, fecha_asig)

- FK *para atender a*: cuit_clie → CLIENTE
- FK *asignación de*: legajo_vend → VENDEDOR
- **Precio_referencia**(codigo_prod,fecha_vigencia,valor)
FK *dado a*: codigo_prod → PRODUCTO
- **Precio_convenio**(codigo_prod,cuit_clie,fecha_vigencia,valor)
FK *dado a*: codigo_prod → PRODUCTO
FK *acordado con*: cuit_clie → CLIENTE
- **Pedido**(nro_pedi,fecha_pedi,cuit_clie,legajo_ven,codigo_prod,cantidad)
FK *realizado por*: cuit_clie → CLIENTE
FK *tomado por*: legajo_vend → VENDEDOR
FK *correspondiente a*: codigo_prod → PRODUCTO
- **Entrega**(nro_entr,cuit_clie,fecha_entr)
FK *entregada a*: cuit_clie → CLIENTE
- **Entrega_pedido**(nro_entr,nro_pedi,cantidad)
FK *línea de*: nro_entr → ENTREGA
FK *correspondiente a*: nro_pedi → PEDIDO

Resolver las siguientes consultas en **SQL**:

- a. Obtener el número de aquellos pedidos que no hayan sido totalmente entregados y el nombre del cliente que realizo el pedido.
Observación: como la cantidad entregada puede ser menor que la pedida, un pedido puede satisfacerse totalmente en varias entregas.
- b. Obtener el CUIT y el nombre de todos los clientes que al día de hoy tengan precios convenidos para todos los productos y que, además, el precio convenido para cada uno de ellos sea menor al precio de referencia.

13.

Dada la siguiente BD de recetas médicas:

- **ELABORA**(laboratorio,droga)
- **PRODUCE**(laboratorio,farmaco)
- **VENDE**(farmacia,farmaco, precio)
- **COMPUESTO_POR**(farmaco,droga)
- **RECETA**(receta_id, fecha, medico, paciente)
- **RECETA_FARMACO**(receta_id, farmaco)

ELABORA indica las drogas que elabora cada laboratorio. PRODUCE indica los fármacos que produce cada laboratorio. VENDE indica los fármacos que vende cada farmacia y su precio correspondiente (un mismo fármaco puede tener diferentes precios en distintas farmacias). COMPUESTO_POR indica las drogas que componen cada fármaco. RECETA tiene la cabecera de cada receta confeccionada. RECETA_FARMACO tiene los fármacos prescritos en cada receta.

- a. Expresar en **AR** y **CRT** las siguientes consultas:
 - I. Listar los laboratorios que no elaboran ninguna de las drogas de los fármacos que producen.
 - II. Listar las drogas que son componentes de todos los fármacos que fueron prescritos al menos dos veces antes del “01/10/2010”.
- b. Expresar en **SQL** las siguientes consultas:

- I. Listar las farmacias que venden todos los fármacos que fueron prescritos más de 5 veces después del “30/09/2010”.
- II. Listar los fármacos, cuyos precios promedio son mayores a los de todos los prescritos por los médicos que más fármacos recetaron.
- III. Obtener los nombres de los fármacos que son equivalentes (en forma de pares, fármaco1, fármaco2), es decir que están compuestos por las mismas drogas.

14.

Dada la siguiente BD con los partidos de fútbol de Primera A:

- **PARTIDO** (equipo1, equipo2, torneo, goles1, goles2, puntos1, puntos2)
 - **JUGADOR** (nombre, edad, equipo)
- Cada partido se registra una sola vez. Por ejemplo si Boca le ganó a River en el torneo “Apertura 2002” por 2 a 1, se registra únicamente la tupla (“Boca”, “River”, “Apertura 2002”, 2, 1, 3, 0)
 - Los candidatos a campeones de un torneo se determinan sumando los puntos obtenidos por cada equipo, seleccionando los que suman el máximo valor. Al haber más de un candidato, se resuelve el campeonato tomando en cuenta la diferencia de goles.
 - Para simplificar se asume que ningún jugador cambió de equipo y los nombres no se repiten.
- a. Expresar en **AR** las siguientes consultas:
- I. Listar los equipos que jugaron exactamente en dos torneos.
 - II. Listar el jugador más joven y el más viejo de cada uno de los equipos que finalizaron invictos (sin perder ningún partido) en al menos dos torneos – Las tuplas del resultado son de la forma (equipo, nombre1, nombre2)
- b. Dada la misma BD del ejercicio anterior pero ahora considerando que cada partido se registra dos veces. Por ejemplo si Boca le ganó a River en el torneo “Apertura 2002” por 2 a 1, se registran dos tuplas:

(“Boca”, “River”, “Apertura 2002”, 2, 1, 3, 0)
(“River”, “Boca”, “Apertura 2002”, 1, 2, 0, 3)

Expresar en **SQL** las siguientes consultas:

- I. Listar los equipos que hayan finalizado invictos en al menos dos torneos.
- II. Listar los torneos que finalizaron con más de un candidato a campeón.

15.

- a. Dado $R(a)$ un esquema de relación. Expresar en AR la consulta que devuelve las tuplas $t \in r(R)$ tal que $t[a] \leq t'[a] \forall t' \in r(R)$ (o sea, el mínimo a para todas las tuplas de la relación).
- b. Dado $R(a, b)$ un esquema de relación. Expresar en AR la consulta que devuelve las tuplas $t \in r(R)$ tal que $t[b] \leq t'[b] \forall t' \in r(R) / t'[a] = t[a]$ (o sea, los mínimos b por cada uno de los grupos de tuplas de la relación que cumplen que el valor para el atributo a es el mismo).

16.

- a. Dado $R(a,b)$ un esquema de relación. Expresar en AR la consulta que devuelve las tuplas t en $r(R)$ tal que $\exists t, t' \in r(R) / t[a] = t'[a] \text{ y } t[b] \neq t'[b]$.
- b. Dado $R(a,b)$ un esquema de relación. Expresar en AR la consulta que devuelve las tuplas t en $r(R)$ tal que $\exists t, t' \in r(R) / t[a] = t'[a] \text{ y } t[b] \neq t'[b]$, y tal que $\neg \exists t'' / t[a] = t''[a] \text{ y } t[b] \neq t''[b] \text{ y } t[b] \neq t''[b]$.

17.

Dadas R y S dos esquemas de relación de aridad r y s respectivamente, donde $r > s$.

- a. Expresar el cociente entre R y S utilizando los operadores básicos de álgebra relacional.
- b. ¿Cómo expresaría el cociente de R y S utilizando SQL?

Ejercicios Adicionales

Utilizando la base de datos definida en el ejercicio 7 resolver los siguientes ítems

- a. Escribir en SQL embebido las sentencias necesarias para determinar en qué auditorio se puede realizar un determinado evento y actualizar como corresponda la relación EVENTO.
- b. Escribir en SQL embebido las sentencias necesarias para registrar que el miembro $c_miembro$ va a participar de la organización del evento c_evento . Qué consideraciones hay que tener en cuenta?