

Práctico 4

SQL

Objetivos

- Expresar consultas sobre el modelo relacional, usando SQL
- Analizar cómo se construye la consulta o solución usando este lenguaje de consultas

Ejercicio 1.

Sean las siguientes tablas pertenecientes a una base de datos:

Proveedores(numProv, nomProv, categoria, ciudadProv)

Contiene información referente a Proveedores

Productos (numProd, nomProd, color, peso, ciudadProd)

Contiene información referente a las Productos

Proyectos (numProy, nomProy, ciudadProy)

Contiene información referente a los Proyectos

Proveen (numProv, numProd, numProy, cantidad)

Contiene la información de los Proveedores que proveen productos a proyectos (cantidad).

En esta base de datos no hay tablas vacías y se cumplen las siguientes restricciones de inclusión:

$$\begin{aligned}\Pi_{numProv}(Proveen) &\subseteq \Pi_{numProv}(Proveedores) \\ \Pi_{numProd}(Proveen) &\subseteq \Pi_{numProd}(Productos) \\ \Pi_{numProy}(Proveen) &\subseteq \Pi_{numProy}(Proyectos)\end{aligned}$$

Resolver las siguientes consultas en SQL:

- Obtener los números de los proveedores que proveen al proyecto número 1 o al proyecto número 2.
 - Obtener todos los datos de los productos de color “rojo”.
 - Obtener el nombre de los proyectos de ciudades que comienzan con ‘M’.
 - Obtener las distintas categorías de proveedores.
- Dar todas las ternas (ciudad, numProd, ciudad), donde un proveedor de la primera ciudad provee con el producto especificado a un proyecto de la segunda ciudad, con la condición de que las ciudades no deben ser la misma.
 - Obtener todos los números de producto tales que no hay otro producto con peso menor.
 - Dar los nombres de los proveedores que sólo proveen un producto y que proveen dicho producto al menos a un proyecto.
 - Dar los nombres de los proveedores que proveen el mismo producto a todos los proyectos.
- Dar la cantidad de productos provistos por cada proveedor.
 - Dar el proyecto que es provisto por más proveedores.
 - Dar los productos que son provistos a al menos 10 proyectos.

Ejercicio 2.

Sea una base de datos de una compañía aérea con las siguientes tablas:

VUELOS (nroVuelo, ciudadOrigen, ciudadDestino, horaSal, horaLleg, distancia)

Corresponde con los vuelos que puede realizar la compañía.

AVION (nroAvion, tipoAvion, horasVuelo)

Corresponde a los datos de cada avión de la compañía.

PILOTOS (nroPiloto, nombre)

Corresponde a los datos de cada uno de los pilotos que trabaja en la compañía.

VIAJES (nroVuelo, fecha, nroAvion, nroPiloto)

Corresponde con los vuelos que efectivamente se realizan y la asignación del avión y el piloto al mismo.

En esta base de datos no hay tablas vacías y se cumplen las siguientes restricciones de inclusión:

$$\Pi_{nroVuelo}(VIAJES) \subseteq \Pi_{nroVuelo}(VUELOS)$$

$$\Pi_{nroAvion}(VIAJES) \subseteq \Pi_{nroAvion}(AVION)$$

$$\Pi_{nroPiloto}(VIAJES) \subseteq \Pi_{nroPiloto}(PILOTOS)$$

Resolver en SQL las siguientes consultas:

- Cantidad de vuelos realizados por cada piloto.
- Ciudad de destino a la que llegaron vuelos realizados por pilotos con más de 10 vuelos.
- Número de los pilotos que pilotearon todos los aviones del tipo "Boeing 737" de la compañía.
- Número de los pilotos con más de 30 vuelos que pilotearon sólo aviones de tipo "DC-10".
- Número de vuelo de los vuelos con distancia mayor a 7000 km que nunca fueron realizados por pilotos con 20 vuelos o más.
- Parejas número de piloto, número de avión tal que todos los vuelos realizados por el piloto fueron en ese avión.

Ejercicio 3.

Una empresa de venta de pinturas con varias sucursales decide formar una sección que se encargue de realizar las combinaciones de colores requeridas por los clientes. Allí se mantiene una base de datos con los pedidos para poder obtener estadísticas acerca de las mezclas más frecuentes, la cual posee las siguientes tablas:

Colores (idC, descripcion, tipo, brillo)

Representa la información de cada uno de los colores, donde *descripcion* es el nombre asignado al color, *tipo* indica si el color es puro (provisto por los fabricantes, sin necesidad de realizar mezclas) o mezcla (ese color no viene preparado de fábrica, hay que realizar la mezcla), *brillo* indica si el color es brillante o mate. El atributo descripción también identifica al color.

Mezclas (idC, idCPuro, proporcion)

Representa la información de los colores que se obtienen como combinación de colores puros. *IdC* es el identificador del color mezcla, e *idCPuro* y *proporción* indican la proporción del color puro para realizar la mezcla.

Productos (IdP, descripcion, marca, tamaño)

Representa la información de los productos (latas de pintura). *Descripcion* es el nombre del producto, *marca* es la marca del producto y *tamaño* es la cantidad de litros de la lata. Se consideran productos distintos los que tienen diferentes tamaños, pero no los que tienen diferentes colores, es decir, que *IdP* no indica el color del producto.

ColorProd (idP, idC)

Representa la información de los colores puros en que viene cada producto. *IdP* es el identificador del producto, e *idC* es el identificador del color. Sólo se venden latas de colores puros.

Solicitudes (idSol, fecha, sucursal, idC, tamaño)

Representa las solicitudes de mezclas realizadas por las diferentes sucursales. Sólo se solicitan mezclas, no colores puros. *IdSol* es un identificador, *fecha* es la fecha en que se realizó la solicitud, y *sucursal* la sucursal de la empresa desde la que se hizo la misma. *IdC* es el identificador de color y tamaño es la cantidad de litros de la mezcla pedida.

En esta base de datos no hay tablas vacías y se cumplen las siguientes restricciones de inclusión:

Resolver en SQL las siguientes consultas.

- Dar las marcas, que entre sus productos, tienen todos los colores necesarios para realizar todas las mezclas.
- Dar el color puro utilizado en más de 3 mezclas (Observación: puede ser más de uno).
- Dar la lista de mezclas mate solicitadas más de 5 veces, y la cantidad total de litros solicitados de cada una.
- Dar la lista de colores que pueden prepararse mezclando, únicamente, colores del producto P13.

Ejercicio 4.

Se modela el sistema de control de asignaciones de tareas a secciones y empleados de una empresa mediante los siguientes esquemas de relación:

FUNCIONARIOS (nroF, nomF, aIng)

Representa la información de todos los empleados de la empresa. Donde *nroF* representa el número de funcionario, *nomF* el nombre del funcionario y *aIng* representa el año de ingreso del funcionario a la empresa.

SECCIONES (codS, nomS, local, tel)

Representa la información de todas las secciones que posee la empresa. Donde *codS* representa el código de la sección, *nomS* su nombre, *tel* número de teléfono de la sección y *local* el local donde esta ubicada. En un local puede haber varias secciones.

TAREAS (codT, nomT, duración) Representa la información de todas las tareas de la empresa. Donde *codT* representa el código de la tarea, *nomT* su nombre y *duración* la duración estimada en días de la misma.

PREVIAS (codTP, codTS) Representa la relación de orden que existe entre las tareas.

Una tupla (*p*,*s*) de esta relación, significa que la tarea con código *p* debe realizarse inmediatamente antes que la tarea con código *s*.

TRABAJA (nroF, codS, hsSem, sueldo, cargo) Representa la cantidad de horas semanales, el cargo que tiene y el sueldo que gana cada funcionario en cada sección en que trabaja.

ASIGNACION (nroF, codT, fAsig)

Representa las asignaciones de tareas realizadas.

Una tupla (*fun*, *tar*, *fe*) de esta relación representa que al funcionario de número "fun" se le ha asignado la tarea con código "tar" en la fecha "fe".

OBLIGACIONES (codS, codT) Representa la relación de obligatoriedad que existe entre las secciones y algunas tareas que deben realizar.

Una tupla (sec, tar) de esta relación representa que la sección con código "sec" está obligada a realizar la tarea con código "tar".

$$\begin{aligned} \Pi_{codTP}(PREVIAS) &\subseteq \Pi_{codT}(TAREAS) \\ \Pi_{codTS}(PREVIAS) &\subseteq \Pi_{codT}(TAREAS) \\ \Pi_{nroF}(TRABAJA) &\subseteq \Pi_{nroF}(FUNCIONARIOS) \\ \Pi_{codS}(TRABAJA) &\subseteq \Pi_{codS}(SECCIONES) \\ \Pi_{nroF}(ASIGNACION) &\subseteq \Pi_{nroF}(FUNCIONARIOS) \\ \Pi_{codT}(ASIGNACION) &\subseteq \Pi_{codT}(TAREAS) \\ \Pi_{codS}(OBLIGACIONES) &\subseteq \Pi_{codS}(SECCIONES) \\ \Pi_{codT}(OBLIGACIONES) &\subseteq \Pi_{codT}(TAREAS) \end{aligned}$$

Resuelva las siguientes consultas en SQL

- Obtener el código, el nombre y la duración de las tareas que se deben realizar inmediatamente antes de la tarea con código "c42".
- Obtener los nombres y año de ingreso de los funcionarios que tienen algún sueldo mayor que algún sueldo del funcionario con número 812.
- Obtener las parejas de números de funcionarios tales que el primero tiene asignada una tarea que debe realizarse inmediatamente antes que una tarea asignada al segundo funcionario.
- Obtener los nombres de los funcionarios tales que en la misma fecha le asignaron más de una tarea.
- Obtener los datos de las tareas inmediatamente previas e inmediatamente sucesoras de las tareas de mayor duración.
- Hallar los nombres de los funcionarios que ingresaron antes del año 1986 y sólo fueron asignados a tareas cuya duración no supera los 60 días.
- Obtener los datos de las secciones, tal que cada sección está obligada a realizar todas las tareas con nombre "URGENTE1".
- Obtener los sueldos totales de los funcionarios que tienen la mayor carga horaria total.
- Obtener los datos de las tareas asignadas a funcionarios que trabajan en secciones que están en locales donde hay un sólo teléfono interno para todo el local.
- Obtener los locales donde trabajan la menor cantidad de funcionarios.

Ejercicio 5.

Sea una base de datos de una compañía de construcciones con las siguientes tablas:

OBRAS (nroObra, director, tipoObra, fechaComienzo)

TRABAJA (nroObrero, nroObra, fecha, codTarea)

PERSONAL (nroObrero, nomObrero, especialidad)

TAREAS (codTarea, descripción, duración)

En esta base de datos se cumplen las siguientes restricciones de inclusión y no hay tablas vacías:

$$\begin{aligned} \{t.nroObrero / TRABAJA(t)\} &\subseteq \{p.nroObrero / PERSONAL(p)\} \\ \{t.nroObra / TRABAJA(t)\} &\subseteq \{o.nroObra / OBRAS(o)\} \end{aligned}$$

$$\{t.codTarea / TRABAJA(t)\} \subseteq \{v.codTarea / TAREAS(v)\}$$

Resolver las siguientes consultas en SQL:

- (a) Dar las ternas (nroObra, codTarea, nroObrero) tales que la tarea corresponde a la obra y el obrero trabajó en la obra pero no realizó (en dicha obra) la tarea mencionada.
- (b) Dar los nombres de los obreros que trabajaron en todas las obras que tienen como director a "AL GUT".
- (c) Dar las parejas de números de obreros que siempre realizaron las mismas tareas en todas las obras donde trabajaron juntos.
- (d) Nombre de los obreros de especialidad "CARPINTERIA" que solo realizaron tareas con duración menor a 5 en todas las obras de tipo "RECONSTRUCCIÓN".
- (e) Dar las ternas (nroObra, cantTareas, cantObreros) tales que para cada número de obra se indica la cantidad de tareas distintas que se realizan y la cantidad de obreros distintos que trabajan en la misma.
- (f) Dar las ternas (nroObra, especialidad, cantObreros) donde para cada obrero se indica la cantidad de obreros de cada una de las especialidades que han trabajado en la obra.
- (g) Obtener la fecha de comienzo de las obras en las que se realizan más de 50 tareas distintas.
- (h) Obtener los códigos de las tareas y su descripción para las tareas que tienen una duración mayor o igual que el promedio general de duración de las tareas.
- (i) Obtener el nombre de los obreros que realizaron más de 5 tareas que tienen una duración mayor o igual que el promedio general de duración de las tareas.
- (j) Obtener el tipo de las obras donde trabajan la mayor cantidad de obreros.