

PROYECTO BASE DE DATOS I

En la fase 2 de nuestro proyecto del curso Base de datos I implementamos la base de datos Cinemark creada para la primera fase del proyecto, realizando las consultas que se nos exigían y llevando el mejor control posible en la manipulación de la información.

José
Orellana
1065918
Alexander
Villatoro
1182118

Proyecto Base de Datos I

Sistema de consultas (Fase 2)

Consultas:

--1. La capacidad total de cada cine, así como el número de salas que tienen cada --tipo de soporte.

```
select Cine.Nombre as NombreCine, SUM(CONVERT(int, Sala.Capacidad)) as
CapacidadTotal,
count(Sala.[No.Sala]) as NumeroDeSalas from Sala inner join Cine
on Cine.IdCine = Sala.IdCine group by Cine.Nombre
```

--2. El mes que hubo más bajas de personal, por cine.

```
select ReporteBajas.IdCine, MONTH(ReporteBajas.Fecha) as Mes,
count(ReporteBajas.CodigoEmpleado) as
CantBajas from ReporteBajas group by ReporteBajas.IdCine, MONTH(ReporteBajas.Fecha)
```

--3. El detalle del actor que ha aparecido en más películas regulares.

```
select top 1 Actor.NombreArtistico as Nombre, Actor.Nacionalidad as Nacionalidad,
count(Actor.NombreArtistico) as CantPeliculas
from Actor inner join Reparto on Actor.NombreArtistico = Reparto.NombreActor
group by Actor.NombreArtistico, Actor.Nacionalidad order by CantPeliculas desc
```

--4. El número de películas calificadas por los asistentes por calificación general
--(Obra Maestra, Muy Buena, Buena, Regular, Mala)

```
select count(Opinion.Calificacion) as NumeroPeliculas from Opinion
```

--5. Los detalles del cine con más bajas de personal.

```
select top 1 Cine.IdCine, Cine.Nombre, Cine.Telefono, Cine.Direccion,
count(ReporteBajas.CodigoEmpleado) as NumeroBajas
from Cine inner join ReporteBajas on Cine.IdCine = ReporteBajas.IdCine
group by Cine.IdCine, Cine.Nombre, Cine.Direccion, Cine.Telefono
order by NumeroBajas desc
```

--6. La cartelera de la semana corriente de cada cine

```
select Cine.Nombre as NombreCine, Cartelera.IdContenido, Cartelera.[No.Sala],
Cartelera.FechaExhibicion
from Cartelera inner join Sala on Cartelera.[No.Sala] = Sala.[No.Sala] inner join
Cine on Cine.IdCine = Sala.IdCine where
cast(DAY(Cartelera.FechaExhibicion) as int) between '06' AND '10'
```

--7. El detalle de las películas regulares que se han exhibido en la sala más
pequeña

--de cada cine. --REVISAR

```
select top 1 PeliculaRegular.IdPelicula, PeliculaRegular.IdContenido,
Sala.[No.Sala], Contenido.Titulo,
Sala.Capacidad as CapacidadSala from PeliculaRegular
inner join Contenido on PeliculaRegular.IdContenido = Contenido.IdContenido inner
join
Cartelera on Contenido.IdContenido = Cartelera.IdContenido inner join Sala
on Cartelera.[No.Sala] = Sala.[No.Sala] order by CONVERT(int,Sala.Capacidad) asc
```

--8. El detalle del cine con mayores ventas en dulcería que en boletos.

```
CREATE TABLE #VentasProducto (NombreCine nvarchar(50), VentasProduct numeric(8,2))
```

```

insert into #VentasProducto
select Cine.Nombre, SUM(ProductoDulceria.Precio) as VentaProducto from Cine inner
join
Compra on Cine.IdCine = Compra.IdCine inner join ProductoDulceria on Compra.IDCompra
=
ProductoDulceria.IDCompra
group by Cine.Nombre order by VentaProducto desc

select #VentasProducto.NombreCine, #VentasProducto.VentasProduct as VentasProducto,
sum(Boleto.Precio) as VentasBoleto from #VentasProducto inner join
Cine on #VentasProducto.NombreCine = Cine.Nombre inner join Boleto on
Cine.IdCine = Boleto.IdCine group by #VentasProducto.VentasProduct,
#VentasProducto.NombreCine
having (#VentasProducto.VentasProduct > sum(Boleto.Precio))

drop table #VentasProducto

--ProductoDulceria.Precio as VentaProducto
--9. El listado de actores que aparecen en la película mejor evaluada.
CREATE TABLE #MejorPelicula (Titulo nvarchar(50), NumeroOpinion int)
insert into #MejorPelicula
select top 1 Contenido.Titulo, count(Opinion.Calificacion) as NumeroOpinion
from Contenido inner join Opinion on
Contenido.IdContenido = Opinion.IdContenido where (Opinion.Calificacion = 'Obra
Maestra')
group by Contenido.Titulo order by NumeroOpinion desc

select Reparto.NombreActor, #MejorPelicula.Titulo, #MejorPelicula.NumeroOpinion
from #MejorPelicula inner join Contenido on #MejorPelicula.Titulo = Contenido.Titulo
inner join PeliculaRegular on Contenido.IdContenido = PeliculaRegular.IdContenido
inner join Reparto on PeliculaRegular.IdPelicula = Reparto.IdPelicula

drop table #MejorPelicula

--10. El detalle de los cines donde se exhibieron las 3 películas peor evaluadas.
select top 3 Cine.IdCine, Cine.Nombre, Cine.Direccion, Cine.Telefono,
Contenido.Titulo,
Opinion.Calificacion
from Cine inner join Sala on Cine.IdCine = Sala.IdCine inner join Evento
on Sala.[No.Sala] = Evento.[No.Sala] inner join Contenido
on Evento.IdContenido = Contenido.IdContenido inner join Opinion
on Contenido.IdContenido = Opinion.IdContenido where (Opinion.Calificacion =
'Mala')

--11. Los datos de las 10 películas de las que más boletos se han emitido.
select top 10 Contenido.Titulo, Contenido.Genero, Contenido.TipoFormato,
Contenido.Duracion
,Contenido.Distribuidora, Contenido.Director, Contenido.Año,
Contenido.PaisProduccion, Contenido.EdadRecomendada
,Contenido.Descripción, Count(Boleto.IDCompra) as NumeroBoletos from Contenido inner
join Evento
on Contenido.IdContenido = Evento.IdContenido inner join Boleto
on Boleto.IdEvento = Evento.IdEvento group by Contenido.Titulo, Contenido.Genero,
Contenido.TipoFormato, Contenido.Duracion, Contenido.Distribuidora,
Contenido.Director, Contenido.Año,
Contenido.PaisProduccion, Contenido.EdadRecomendada, Contenido.Descripción
order by NumeroBoletos desc

```

```

--12. La duración promedio de los contenidos exhibidos por cine.
select Cine.Nombre, AVG(CONVERT(int,substring (Contenido.Duracion, 1, 1))) as
PromedioDuracion from Contenido
inner join Cartelera on Contenido.IdContenido = Cartelera.IdContenido inner join
Sala on Cartelera.[No.Sala] = Sala.[No.Sala] inner join Cine on Sala.IdCine =
Cine.IdCine group by Cine.Nombre

--13. El detalle del cliente que más visitas a Cinemark ha realizado, indicando el
--número de veces que ha visitado cada sede.
CREATE TABLE #ClienteFrecuente (NIT nvarchar(8), Nombre nvarchar(50), Direccion
nvarchar(200),
Visitas int)
insert into #ClienteFrecuente
select top 1 Cliente.NIT, Cliente.Nombre, Cliente.Direccion, count(Cine.Nombre) as
Visitas from
Cliente inner join Factura on Cliente.NIT = Factura.NIT inner join Compra on
Factura.IDCompra = Compra.IDCompra inner join Cine on Compra.IdCine = Cine.IdCine
group by Cliente.NIT, Cliente.Nombre, Cliente.Direccion order by Visitas desc

select #ClienteFrecuente.NIT, #ClienteFrecuente.Nombre, #ClienteFrecuente.Direccion,
Cine.Nombre as
NombreCine, count(Cine.Nombre) as NumVisitas from #ClienteFrecuente inner join
Factura on
#ClienteFrecuente.NIT = Factura.NIT inner join Compra on Factura.IDCompra =
Compra.IDCompra
inner join Cine on Compra.IdCine = Cine.IdCine group by #ClienteFrecuente.NIT,
#ClienteFrecuente.Nombre,
#ClienteFrecuente.Direccion, Cine.Nombre, Cine.Nombre

drop table #ClienteFrecuente

--14. Los dos productos más populares en dulcería por cine.
CREATE TABLE #ComprasCine (Nombre nvarchar(50), Producto nvarchar(50), CantProd int)
insert into #ComprasCine
select Cine.Nombre, ProductoDulceria.Producto, count(ProductoDulceria.Producto) as
CantProd from
ProductoDulceria inner join Compra on ProductoDulceria.IDCompra = Compra.IDCompra
inner join
Cine on Compra.IdCine = Cine.IdCine group by Cine.Nombre, ProductoDulceria.Producto
order by CantProd desc

select #ComprasCine.Nombre, #ComprasCine.Producto, MAX(#ComprasCine.CantProd) as
Cantidad from #ComprasCine
group by #ComprasCine.Nombre, #ComprasCine.Producto

drop table #ComprasCine

--15. El promedio de ventas totales por cine
CREATE TABLE ##VentasBoleto (Nombre nvarchar(50), VentasProductos numeric(8,2))
insert into ##VentasBoleto
select Cine.Nombre, sum(Boleto.Precio) as VentasBoletos
from Compra inner join Boleto on Compra.IDCompra = Boleto.IDCompra inner join Cine
on Compra.IdCine = Cine.IdCine group by Cine.Nombre

select Cine.Nombre, (##VentasBoleto.VentasProductos + sum(ProductoDulceria.Precio))
/ 2 as
PromVentasTotales from Compra inner join ProductoDulceria on Compra.IDCompra =
ProductoDulceria.IDCompra inner join Cine on Compra.IdCine = Cine.IdCine inner join

```

```
##VentasBoleto on Cine.Nombre = ##VentasBoleto.Nombre group by Cine.Nombre,  
##VentasBoleto.VentasProductos
```

```
--16. El listado contenidos que se han dejado de exhibir desde el primero de marzo  
en  
--todos los cines.
```

```
select RegistroHistorico.IdContenido, Contenido.Titulo from RegistroHistorico inner  
join Contenido on  
RegistroHistorico.IdContenido = Contenido.IdContenido where  
(MONTH(RegistroHistorico.FechaUltimaExhibición) >= '03')
```

```
--17. El nombre de los empleados que han laborado en el cine con mayores ventas.
```

```
CREATE TABLE #CineMasVentas (Nombre nvarchar(50), VentasTotal numeric(8,2))  
insert into #CineMasVentas  
select top 1 Cine.Nombre, (##VentasBoleto.VentasProductos +  
sum(ProductoDulceria.Precio)) as  
VentasTotales from Compra inner join ProductoDulceria on Compra.IDCompra =  
ProductoDulceria.IDCompra inner join Cine on Compra.IdCine = Cine.IdCine inner join  
##VentasBoleto on Cine.Nombre = ##VentasBoleto.Nombre  
group by Cine.Nombre, ##VentasBoleto.VentasProductos  
order by VentasTotales desc
```

```
select #CineMasVentas.Nombre, Empleado.CodigoEmpleado,  
Empleado.Nombre, #CineMasVentas.VentasTotal  
from #CineMasVentas inner join Cine on #CineMasVentas.Nombre =  
Cine.Nombre inner join RegistroRotacion on Cine.IdCine = RegistroRotacion.IdCine  
inner join Empleado on RegistroRotacion.CodigoEmpleado = Empleado.CodigoEmpleado
```

```
drop table #CineMasVentas  
drop table ##VentasBoleto
```

```
--18. El número de exhibiciones culturales y deportivas que se han proyectado por  
cine.
```

```
CREATE TABLE #Deportiva (NombreCine nvarchar(50), CantidadDep int)  
insert into #Deportiva  
Select Cine.Nombre as NombreCine, Count(AcontecimientoDeportivo.IdExhibicionE) as  
CantidadDep  
from AcontecimientoDeportivo inner join ExhibicionEspecial  
on AcontecimientoDeportivo.IdExhibicionE = ExhibicionEspecial.IdExhibicionE  
inner join Contenido on ExhibicionEspecial.IdContenido = Contenido.IdContenido  
inner join Evento on Contenido.IdContenido = Evento.IdContenido  
inner join Sala on Evento.[No.Sala] = Sala.[No.Sala]  
inner join Cine on Sala.IdCine = Cine.IdCine  
group by Cine.Nombre
```

```
select #Deportiva.NombreCine, #Deportiva.CantidadDep,  
count(AcontecimientoCultural.IdExhibicion)  
as CantidadCul from #Deportiva inner join Cine on #Deportiva.NombreCine =  
Cine.Nombre  
inner join Sala on Cine.IdCine = Sala.IdCine inner join Evento on Sala.[No.Sala] =  
Evento.[No.Sala] inner join Contenido on Evento.IdContenido = Contenido.IdContenido  
inner join ExhibicionEspecial on Contenido.IdContenido =  
ExhibicionEspecial.IdContenido  
full outer join AcontecimientoCultural on ExhibicionEspecial.IdExhibicionE =  
AcontecimientoCultural.IdExhibicion  
group by #Deportiva.NombreCine, #Deportiva.CantidadDep
```

```

drop table #Deportiva
--19. El mes que menos boletos se vendieron, pero más productos de dulcería se
vendieron.
CREATE TABLE #MesBoleto (Mes int, CantBoleto int)
insert into #MesBoleto
select Month(Factura.Fecha) as Mes, count(Boleto.IDCompra) as CantBoleto from Boleto
inner join Compra
on Boleto.IDCompra = Compra.IDCompra inner join Factura on Compra.IDCompra =
Factura.IDCompra
group by Month(Factura.Fecha)

select top 1 #MesBoleto.Mes, #MesBoleto.CantBoleto, count(ProductoDulceria.IDCompra)
as CantProducto from
ProductoDulceria inner join Compra on ProductoDulceria.IDCompra = Compra.IDCompra
inner join
Factura on Compra.IDCompra = Factura.IDCompra inner join #MesBoleto on
#MesBoleto.Mes =
month(Factura.Fecha) group by #MesBoleto.Mes, #MesBoleto.CantBoleto
having (#MesBoleto.CantBoleto < count(ProductoDulceria.IDCompra))

drop table #MesBoleto

--20. El detalle de los clientes que nunca han consumido algo de las dulcerías en
--alguna de sus visitas.
select Cliente.NIT, Cliente.Nombre, Cliente.Direccion,
count(ProductoDulceria.Producto) as ProdDulceria
from Factura inner join Cliente
on Factura.NIT = Cliente.NIT inner join Compra on Factura.IDCompra = Compra.IDCompra
left join
ProductoDulceria on Compra.IDCompra = ProductoDulceria.IDCompra
group by Cliente.NIT, Cliente.Nombre, Cliente.Direccion
having(count(ProductoDulceria.Producto) = 0)

```

Selección de vistas

Al momento de crear una vista es mucho más sencilla la manipulación de datos mediante el query ya que el usuario que haga las consultas no debe preocuparse por algún teclaso de más o de hacer algo que pueda alterar o modificar el query y pueda ser perjudicial al momento de obtener los datos. Es por eso que para crear las vistas nosotros como programadores hemos seleccionado las consultas más importantes o que pueden ser llamadas con más frecuencia para dar información relevante a la empresa o a los usuarios.

- 1) La consulta 1 puede ser una vista ya que nos demuestra la capacidad total de cada cine y el número de salas por cine para que al momento se tenga un control al momento de manejar la venta de boletos. Es algo que se puede consultar frecuentemente.
- 2) La consulta numero 2 puede ser una vista ya que muestra las bajas de cada mes por cada cine y se mantiene un control para que esas bajas ya no se estén dando.
- 3) La consulta numero 6 puede ser una vista ya que esta nos va a dar la información de la cartelera de cada película en su diferente cine, esto es importante para que los Clientes tengan conocimiento de que película está en exhibición.
- 4) La consulta numero 8 podría ser una vista ya que obtenemos la información de los cines en donde los productos de las dulcerías se venden más, esto para saber en qué cine tiene mejores productos que en otros cines.
- 5) La consulta 9 puede ser una vista ya que nos muestra el listado de actores que aparecen en las mejores películas evaluadas, esto es importante ya que algunos clientes tienen actores favoritos y estos compraran películas en donde estén esos actores.
- 6) La consulta numero 11 muestra los datos de las 10 películas en donde se han vendido más boletos así que en esta consulta puede ser una vista, ya que se puede realizar publicidad en base a las películas más vistas para vender más boletos.
- 7) La consulta numero 13 puede ser una vista ya que nos muestra el cliente que más visitas a Cinemark ha hecho, esto puede servir para darle algún tipo de descuento o recompensa por su preferencia.

- 8) La consulta numero 14 puede ser una vista ya que el personal de cada cine sabe qué productos son los más solicitados y de acuerdo a eso producir más.
- 9) La consulta numero 15 puede ser una vista ya que esta nos dice en qué cine las ventas incrementan con más frecuencia.
- 10) La consulta numero 17 puede ser una vista ya que en esta nos van a decir que empleados están haciendo un buen trabajo para que el cine tenga mayores ventas, en cuál cine hay un mejor servicio.