Adv Database Project1-

* **SQL:**

1. DES> select \* from Album where ArtistId in (select ArtistId from Artist where Name like '%Red Hot Chili Peppers%')

Warning: [Sem] Unnecessary DISTINCT because of primary key in [Artist].

answer(Album.AlbumId:int,Album.Title:varchar(160),Album.ArtistId:int) ->

{

answer(193,'Blood Sugar Sex Magik',127),

answer(194,'By The Way',127),

answer(195,'Californication',127)

}

Info: 3 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> select \* from Genre where GenreId in (Select GenreId from Track where AlbumId in (Select AlbumId from Album where ArtistId in (Select ArtistId from Artist where Name = 'U2' ) ))

Warning: [Sem] Unnecessary DISTINCT because of primary key in [Album].

Warning: [Sem] Unnecessary DISTINCT because of primary key in [Artist].

answer(Genre.GenreId:int,Genre.Name:varchar(120)) ->

{

answer(1,'Rock'),

answer(9,'Pop')

}

Info: 2 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> Select al.Title, ar.Name from Album al inner join Artist ar on ar.ArtistId=al.ArtistId where AlbumId in (Select AlbumId from Track where TrackId in (Select TrackId from PlaylistTrack where PlaylistId in (Select PlaylistId from Playlist where Name = 'Grunge')))

Warning: [Sem] Unnecessary DISTINCT because of primary key in [Playlist].

answer(al.Title:varchar(160),ar.Name:varchar(120)) ->

{

answer('A-Sides','Soundgarden'),

answer('Core','Stone Temple Pilots'),

answer('Facelift','Alice In Chains'),

answer('Nevermind','Nirvana'),

answer('Temple of the Dog','Temple of the Dog'),

answer('Ten','Pearl Jam'),

answer('Vs.','Pearl Jam')

}

Info: 7 tuples computed.

---------------------------------------------------------------------------------------------------------------------

4.a DES> select FirstName,LastName,Email from Customer where CustomerId in (Select CustomerId from Invoice where InvoiceId in (Select InvoiceId from InvoiceLine where TrackId in (Select TrackId from PlaylistTrack where PlaylistId in (Select PlaylistId from Playlist where Name = 'TV Shows'))))

b. DES> select ci.FirstName,ci.LastName,ci.Email from Customer ci inner join Invoice inv ON inv.CustomerId=ci.CustomerId inner join InvoiceLine il On il.InvoiceId=inv.InvoiceId inner join PlaylistTrack pt On pt.TrackId=il.TrackId inner join Playlist pl ON pl.PlaylistId=pt.PlaylistId where pl.Name = 'TV Shows'

answer(ci.FirstName:varchar(40),ci.LastName:varchar(20),ci.Email:varchar(60)) ->

{

answer('Astrid','Gruber','astrid.gruber@apple.at'),

answer('Bjørn','Hansen','bjorn.hansen@yahoo.no'),

answer('Camille','Bernard','camille.bernard@yahoo.fr'),

answer('Dan','Miller','dmiller@comcast.com'),

answer('Dominique','Lefebvre','dominiquelefebvre@gmail.com'),

answer('Frank','Ralston','fralston@gmail.com'),

answer('František','Wichterlová','frantisekw@jetbrains.com'),

answer('François','Tremblay','ftremblay@gmail.com'),

answer('Fynn','Zimmermann','fzimmermann@yahoo.de'),

answer('Heather','Leacock','hleacock@gmail.com'),

answer('Helena','Holý','hholy@gmail.com'),

answer('Hugh','O\'Reilly','hughoreilly@apple.ie'),

answer('Isabelle','Mercier','isabelle\_mercier@apple.fr'),

answer('Jack','Smith','jacksmith@microsoft.com'),

answer('Jennifer','Peterson','jenniferp@rogers.ca'),

answer('Joakim','Johansson','joakim.johansson@yahoo.se'),

answer('Johannes','Van der Berg','johavanderberg@yahoo.nl'),

answer('João','Fernandes','jfernandes@yahoo.pt'),

answer('Julia','Barnett','jubarnett@gmail.com'),

answer('Ladislav','Kovács','ladislav\_kovacs@apple.hu'),

answer('Luis','Rojas','luisrojas@yahoo.cl'),

answer('Luís','Gonçalves','luisg@embraer.com.br'),

answer('Manoj','Pareek','manoj.pareek@rediff.com'),

answer('Puja','Srivastava','puja\_srivastava@yahoo.in'),

answer('Richard','Cunningham','ricunningham@hotmail.com'),

answer('Terhi','Hämäläinen','terhi.hamalainen@apple.fi'),

answer('Tim','Goyer','tgoyer@apple.com'),

answer('Victor','Stevens','vstevens@yahoo.com'),

answer('Wyatt','Girard','wyatt.girard@yahoo.fr')

}

Info: 29 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> select FirstName,LastName,Email from Employee where EmployeeId in (select SupportRepId from Customer where CustomerId in (Select CustomerId from Invoice where InvoiceId in (Select InvoiceId from InvoiceLine where TrackId in (Select TrackId from Track where MediaTypeId in (Select MediaTypeId from MediaType where Name = 'Purchased AAC audio file')))))

Warning: [Sem] Unnecessary DISTINCT because of primary key in [Track].

Warning: [Sem] Unnecessary DISTINCT because of primary key in [MediaType].

answer(Employee.FirstName:varchar(20),Employee.LastName:varchar(20),Employee.Email:varchar(60)) ->

{

answer('Margaret','Park','margaret@chinookcorp.com'),

answer('Steve','Johnson','steve@chinookcorp.com')

}

Info: 2 tuples computed.

========================================================================

* **RA:**

1. DES-RA> project Album.Title (select Album.ArtistId=Artist.ArtistId and Artist.Name like '%Red Hot Chili Peppers%' (Album product Artist))

answer(Album.AlbumId:int,Album.Title:varchar(160),Album.ArtistId:int,Artist.ArtistId:int,Artist.Name:varchar(120)) ->

{

answer(193,'Blood Sugar Sex Magik',127,127,'Red Hot Chili Peppers'),

answer(194,'By The Way',127,127,'Red Hot Chili Peppers'),

answer(195,'Californication',127,127,'Red Hot Chili Peppers')

}

Info: 3 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> Project Genre.Name (select Artist.Name='U2' (Genre zjoin Genre.GenreId=Track.GenreId Track zjoin Track.AlbumId=Album.AlbumId Album zjoin Album.ArtistId=Artist.ArtistId Artist))

answer(Genre.Name:varchar(120)) ->

{

answer('Pop'),

answer('Rock')

}

Info: 2 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> project Album.Title,Artist.Name(select Playlist.Name='Grunge' (Artist zjoin Artist.ArtistId=Album.ArtistId Album zjoin Album.AlbumId=Track.AlbumId Track zjoin Track.TrackId=PlaylistTrack.TrackId PlaylistTrack zjoin PlaylistTrack.PlaylistId=Playlist.PlaylistId Playlist))

answer(Album.Title:varchar(160),Artist.Name:varchar(120)) ->

{

answer('A-Sides','Soundgarden'),

answer('Core','Stone Temple Pilots'),

answer('Facelift','Alice In Chains'),

answer('Nevermind','Nirvana'),

answer('Temple of the Dog','Temple of the Dog'),

answer('Ten','Pearl Jam'),

answer('Vs.','Pearl Jam')

}

Info: 7 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES-RA> project Customer.FirstName,Customer.LastName,Customer.Email (Select Playlist.Name='TV Shows'(Customer zjoin Customer.CustomerId=Invoice.CustomerId Invoice zjoin Invoice.InvoiceId=InvoiceLine.InvoiceId InvoiceLine zjoin InvoiceLine.TrackId=PlaylistTrack.TrackId PlaylistTrack zjoin PlaylistTrack.PlaylistId=Playlist.PlaylistId Playlist ))

answer(Customer.FirstName:varchar(40),Customer.LastName:varchar(20),Customer.Email:varchar(60)) ->

{

answer('Astrid','Gruber','astrid.gruber@apple.at'),

answer('BjÃ¸rn','Hansen','bjorn.hansen@yahoo.no'),

answer('Camille','Bernard','camille.bernard@yahoo.fr'),

answer('Dan','Miller','dmiller@comcast.com'),

answer('Dominique','Lefebvre','dominiquelefebvre@gmail.com'),

answer('Frank','Ralston','fralston@gmail.com'),

answer('FrantiÅ¡ek','WichterlovÃ¡','frantisekw@jetbrains.com'),

answer('FranÃ§ois','Tremblay','ftremblay@gmail.com'),

answer('Fynn','Zimmermann','fzimmermann@yahoo.de'),

answer('Heather','Leacock','hleacock@gmail.com'),

answer('Helena','HolÃ½','hholy@gmail.com'),

answer('Hugh','O\'Reilly','hughoreilly@apple.ie'),

answer('Isabelle','Mercier','isabelle\_mercier@apple.fr'),

answer('Jack','Smith','jacksmith@microsoft.com'),

answer('Jennifer','Peterson','jenniferp@rogers.ca'),

answer('Joakim','Johansson','joakim.johansson@yahoo.se'),

answer('Johannes','Van der Berg','johavanderberg@yahoo.nl'),

answer('JoÃ£o','Fernandes','jfernandes@yahoo.pt'),

answer('Julia','Barnett','jubarnett@gmail.com'),

answer('Ladislav','KovÃ¡cs','ladislav\_kovacs@apple.hu'),

answer('Luis','Rojas','luisrojas@yahoo.cl'),

answer('LuÃ­s','GonÃ§alves','luisg@embraer.com.br'),

answer('Manoj','Pareek','manoj.pareek@rediff.com'),

answer('Puja','Srivastava','puja\_srivastava@yahoo.in'),

answer('Richard','Cunningham','ricunningham@hotmail.com'),

answer('Terhi','HÃ¤mÃ¤lÃ¤inen','terhi.hamalainen@apple.fi'),

answer('Tim','Goyer','tgoyer@apple.com'),

answer('Victor','Stevens','vstevens@yahoo.com'),

answer('Wyatt','Girard','wyatt.girard@yahoo.fr')

}

Info: 29 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES-RA> project Employee.FirstName,Employee.LastName,Employee.Email (Select MediaType.Name = 'Purchased AAC audio file'(Employee zjoin Employee.EmployeeId=Customer.SupportRepId Customer zjoin Customer.CustomerId=Invoice.CustomerId Invoice zjoin Invoice.InvoiceId=InvoiceLine.InvoiceId InvoiceLine zjoin InvoiceLine.TrackId=Track.TrackId Track zjoin Track.MediaTypeId=MediaType.MediaTypeId MediaType ))

answer(Employee.FirstName:varchar(20),Employee.LastName:varchar(20),Employee.Email:varchar(60)) ->

{

answer('Margaret','Park','margaret@chinookcorp.com'),

answer('Steve','Johnson','steve@chinookcorp.com')

}

Info: 2 tuples computed.

========================================================================

select Genre.GenreId=track.GenreId and track.AlbumId=Album.AlbumId and Album.ArtistId=Artist.ArtistId and Artist.Name='U2' (Genre product track product Album product Artist)

select \* from Genre where GenreId in (Select GenreId from Track where AlbumId in (Select AlbumId from Album where ArtistId in (Select ArtistId from Artist where Name = 'U2' ) ))

Project Genre.Name (select Artist.Name='U2' (Genre zjoin Genre.GenreId=Track.GenreId Track zjoin Track.AlbumId=Album.AlbumId Album zjoin Album.ArtistId=Artist.ArtistId Artist))

project Name

(project GenreId (Project GenreId ( (Track)))

(Genre))

project Name (Select GenreId like (select GenreId (Select GenreId=25 (Track))) (Genre))

(project Name (Genre)) Product (project Name (Genre Product Genre.GenreId=Track.GenreId Track Product Track.AlbumId=Album.AlbumId Album Product Album.ArtistId=Artist.ArtistId Artist Artist.Name='U2'))

project Artist.name (select Album.ArtistId=Artist.ArtistId and Artist.Name like '%Red Hot Chili Peppers%' (Album product Artist))

========================================================================

* **TRC:**

1. DES> {T.Title |'Album'(T) and (exists R)('Artist'(R) and T.ArtistId=R.ArtistId and R.Name='Red Hot Chili Peppers')}

answer(title:varchar(160)) ->

{

answer('Blood Sugar Sex Magik'),

answer('By The Way'),

answer('Californication')

}

Info: 3 tuples computed.

--------------------------------------------------------------------------------------------------------------------

1. DES-TRC> {S.Name|'Genre'(S) and (Exists T)('Track'(T) and S.GenreId=T.GenreId and (Exists Al)('Album'(Al) and T.AlbumId=Al.AlbumId and (Exists Ar)('Artist'(Ar) and Al.ArtistId=Ar.ArtistId and Ar.Name='U2' )))}

answer(name:varchar(120)) ->

{

answer('Pop'),

answer('Rock')

}

Info: 2 tuples computed.

-------------------------------------------------------------------------------------------------------------------

1. DES> {S.Name, T.Title | 'Artist'(S) and 'Album'(T) and S.ArtistId=T.ArtistId and (exists R)('Track'(R) and T.AlbumId=R.AlbumId and (exists P)('PlaylistTrack'(P) and R.TrackId=P.TrackId and (exists Q)('Playlist'(Q) and P.PlaylistId=Q.PlaylistId and Q.Name='Grunge')))}

answer(name:varchar(120),title:varchar(160)) ->

{

answer('Alice In Chains','Facelift'),

answer('Nirvana','Nevermind'),

answer('Pearl Jam','Ten'),

answer('Pearl Jam','Vs.'),

answer('Soundgarden','A-Sides'),

answer('Stone Temple Pilots','Core'),

answer('Temple of the Dog','Temple of the Dog')

}

Info: 7 tuples computed.

-------------------------------------------------------------------------------------------------------------------

1. DES-TRC> {C.FirstName,C.LastName,C.Email | 'Customer'(C) and (EXISTS I) ('Invoice' (I) and C.CustomerId=I.CustomerId and (Exists IL)('InvoiceLine'(IL) and I.InvoiceId=IL.InvoiceId and (Exists PT)('PlaylistTrack'(PT) and IL.TrackId=PT.TrackId and(Exists PL)('Playlist'(PL) and PT.PlaylistId=PL.PlaylistId and PL.Name='TV Shows')) )) }

answer(firstName:varchar(40),lastName:varchar(20),email:varchar(60)) ->

{

answer('Astrid','Gruber','astrid.gruber@apple.at'),

answer('BjÃ¸rn','Hansen','bjorn.hansen@yahoo.no'),

answer('Camille','Bernard','camille.bernard@yahoo.fr'),

answer('Dan','Miller','dmiller@comcast.com'),

answer('Dominique','Lefebvre','dominiquelefebvre@gmail.com'),

answer('Frank','Ralston','fralston@gmail.com'),

answer('FrantiÅ¡ek','WichterlovÃ¡','frantisekw@jetbrains.com'),

answer('FranÃ§ois','Tremblay','ftremblay@gmail.com'),

answer('Fynn','Zimmermann','fzimmermann@yahoo.de'),

answer('Heather','Leacock','hleacock@gmail.com'),

answer('Helena','HolÃ½','hholy@gmail.com'),

answer('Hugh','O\'Reilly','hughoreilly@apple.ie'),

answer('Isabelle','Mercier','isabelle\_mercier@apple.fr'),

answer('Jack','Smith','jacksmith@microsoft.com'),

answer('Jennifer','Peterson','jenniferp@rogers.ca'),

answer('Joakim','Johansson','joakim.johansson@yahoo.se'),

answer('Johannes','Van der Berg','johavanderberg@yahoo.nl'),

answer('JoÃ£o','Fernandes','jfernandes@yahoo.pt'),

answer('Julia','Barnett','jubarnett@gmail.com'),

answer('Ladislav','KovÃ¡cs','ladislav\_kovacs@apple.hu'),

answer('Luis','Rojas','luisrojas@yahoo.cl'),

answer('LuÃ­s','GonÃ§alves','luisg@embraer.com.br'),

answer('Manoj','Pareek','manoj.pareek@rediff.com'),

answer('Puja','Srivastava','puja\_srivastava@yahoo.in'),

answer('Richard','Cunningham','ricunningham@hotmail.com'),

answer('Terhi','HÃ¤mÃ¤lÃ¤inen','terhi.hamalainen@apple.fi'),

answer('Tim','Goyer','tgoyer@apple.com'),

answer('Victor','Stevens','vstevens@yahoo.com'),

answer('Wyatt','Girard','wyatt.girard@yahoo.fr')

}

Info: 29 tuples computed.

------------------------------------------------------------------------------------------

1. DES-TRC> { E.FirstName,E.LastName,E.Email | 'Employee'(E) and (EXISTS C)('Customer'(C) and E.EmployeeId=C.SupportRepId and (EXISTS I)('Invoice'(I) and C.CustomerId=I.CustomerId and (EXISTS IL)('InvoiceLine'(IL) and I.InvoiceId=IL.InvoiceId and (EXISTS T)('Track'(T) and IL.TrackId=T.TrackId and (EXISTS MT)('MediaType'(MT)and T.MediaTypeId=MT.MediaTypeId and MT.Name= 'Purchased AAC audio file')))))}

answer(firstName:varchar(20),lastName:varchar(20),email:varchar(60)) ->

{

answer('Margaret','Park','margaret@chinookcorp.com'),

answer('Steve','Johnson','steve@chinookcorp.com')

}

Info: 2 tuples computed.

==============================================================================

* **DRC:**

1. DES> {Title | (Exists ArtistId)('Album'( \_,Title ,ArtistId) and 'Artist'(ArtistId,'Red Hot Chili Peppers'))}

answer(title:varchar(160)) ->

{

answer('Blood Sugar Sex Magik'),

answer('By The Way'),

answer('Californication')

}

Info: 3 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> {Name | (Exists GenreId)('Genre'(GenreId,Name) and (Exists AlbumId)('Track'( \_,\_,AlbumId,\_,GenreId,\_,\_,\_,\_) and (Exists ArtistId)('Album'( AlbumId,\_,ArtistId) and 'Artist'(ArtistId,'U2') )))}

(OR)

DES> {Name | (Exists GenreId,AlbumId,ArtistId)('Genre'(GenreId,Name) and 'Track'( \_,\_,AlbumId,\_,GenreId,\_,\_,\_,\_) and 'Album'( AlbumId,\_,ArtistId) and 'Artist'(ArtistId,'U2'))}

answer(name:varchar(120)) ->

{

answer('Pop'),

answer('Rock')

}

Info: 2 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> {Name,Title| (exists ArtistId,AlbumId)('Artist'(ArtistId,Name) and 'Album'(AlbumId,Title,ArtistId ) and (exists TrackId)('Track'( TrackId,\_,AlbumId,\_,\_,\_,\_,\_,\_) and(exists PlaylistId)('PlaylistTrack'(PlaylistId,TrackId) and 'Playlist'(PlaylistId,'Grunge')) ))}

answer(name:varchar(120),title:varchar(160)) ->

{

answer('Alice In Chains','Facelift'),

answer('Nirvana','Nevermind'),

answer('Pearl Jam','Ten'),

answer('Pearl Jam','Vs.'),

answer('Soundgarden','A-Sides'),

answer('Stone Temple Pilots','Core'),

answer('Temple of the Dog','Temple of the Dog')

}

Info: 7 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> {FirstName,LastName,Email | (Exists CustomerId,InvoiceId,TrackId,PlaylistId)('Customer'(CustomerId,FirstName,LastName,\_,\_,\_,\_,\_,\_,\_,\_,Email,\_) and 'Invoice'(InvoiceId,CustomerId,\_,\_,\_,\_,\_,\_,\_) and 'InvoiceLine'(\_,InvoiceId,TrackId,\_,\_) and 'PlaylistTrack'(PlaylistId,TrackId) and 'Playlist'(PlaylistId,'TV Shows') ) }

answer(firstName:varchar(40),lastName:varchar(20),email:varchar(60)) ->

{

answer('Astrid','Gruber','astrid.gruber@apple.at'),

answer('BjÃ¸rn','Hansen','bjorn.hansen@yahoo.no'),

answer('Camille','Bernard','camille.bernard@yahoo.fr'),

answer('Dan','Miller','dmiller@comcast.com'),

answer('Dominique','Lefebvre','dominiquelefebvre@gmail.com'),

answer('Frank','Ralston','fralston@gmail.com'),

answer('FrantiÅ¡ek','WichterlovÃ¡','frantisekw@jetbrains.com'),

answer('FranÃ§ois','Tremblay','ftremblay@gmail.com'),

answer('Fynn','Zimmermann','fzimmermann@yahoo.de'),

answer('Heather','Leacock','hleacock@gmail.com'),

answer('Helena','HolÃ½','hholy@gmail.com'),

answer('Hugh','O\'Reilly','hughoreilly@apple.ie'),

answer('Isabelle','Mercier','isabelle\_mercier@apple.fr'),

answer('Jack','Smith','jacksmith@microsoft.com'),

answer('Jennifer','Peterson','jenniferp@rogers.ca'),

answer('Joakim','Johansson','joakim.johansson@yahoo.se'),

answer('Johannes','Van der Berg','johavanderberg@yahoo.nl'),

answer('JoÃ£o','Fernandes','jfernandes@yahoo.pt'),

answer('Julia','Barnett','jubarnett@gmail.com'),

answer('Ladislav','KovÃ¡cs','ladislav\_kovacs@apple.hu'),

answer('Luis','Rojas','luisrojas@yahoo.cl'),

answer('LuÃ­s','GonÃ§alves','luisg@embraer.com.br'),

answer('Manoj','Pareek','manoj.pareek@rediff.com'),

answer('Puja','Srivastava','puja\_srivastava@yahoo.in'),

answer('Richard','Cunningham','ricunningham@hotmail.com'),

answer('Terhi','HÃ¤mÃ¤lÃ¤inen','terhi.hamalainen@apple.fi'),

answer('Tim','Goyer','tgoyer@apple.com'),

answer('Victor','Stevens','vstevens@yahoo.com'),

answer('Wyatt','Girard','wyatt.girard@yahoo.fr')

}

Info: 29 tuples computed.

---------------------------------------------------------------------------------------------------------------------

1. DES> {FirstName,LastName,Email | (Exists EmployeeId)('Employee'(EmployeeId,LastName,FirstName,\_,\_,\_,\_,\_,\_,\_,\_,\_,\_,\_,Email) and (Exists CustomerId)('Customer'(CustomerId,\_,\_,\_,\_,\_,\_,\_,\_,\_,\_,\_,EmployeeId) and (Exists InvoiceId)('Invoice'(InvoiceId,CustomerId,\_,\_,\_,\_,\_,\_,\_) and (Exists TrackId)('InvoiceLine'(\_,InvoiceId,TrackId,\_,\_) and (Exists MediaTypeId)('Track'(TrackId,\_,\_,MediaTypeId,\_,\_,\_,\_,\_) and ('MediaType'(MediaTypeId,'Purchased AAC audio file')))))))}

answer(firstName:varchar(20),lastName:varchar(20),email:varchar(60)) ->

{

answer('Margaret','Park','margaret@chinookcorp.com'),

answer('Steve','Johnson','steve@chinookcorp.com')

}

Info: 2 tuples computed.

----------------------xxx--------------------