

PRÁCTICA 3. NOSQL, OPTIMIZACIÓN Y TRANSACCIONES

Descripción del nuevo material entregado.

Además del código entregado en la práctica 2 de sistemas informáticos, se han añadido diferentes archivos y códigos:

- **createMongoDBFromPostgreSQLDB.py**: este archivo se encuentra dentro de la carpeta app, en la misma ruta que, por ejemplo, "routes.py" y es el script encargado de crear, a través de su ejecución con "python3 createMongoDBFromPostgreSQLDB.py" la base de datos si1 en mongoDB a partir de la base de datos proporcionada en postgres. Este script también crea la colección "topUSA" que contiene las 800 películas más actuales de Estados Unidos, ordenadas de más a menos actual. También contiene las funciones pedidas en el enunciado (buscar películas con "Life", películas que tengan como director a Woody Allen y películas que hayan participado juntos Parsons y Galecki)
- **Página html de topUsa**: nueva página web con los datos de las consultas mongoDB anteriores almacenadas en una tabla.
- **clientesDistintos.sql**: esta consulta sql muestra el número de clientes distintos que han realizado pedidos que superen un precio en un mes, por ejemplo, que número de clientes han realizado distintos pedidos en enero y que en esos mismos pedidos se hayan gastado más de 1000€.
- **countStatus.sql**: este script sql contiene las consultas, la creación de índices y las sentencias ANALYZE que se desarrollan en el apartado "2) Optimización" de la práctica.
- **Página html de borraCliente**: nueva página html que ejecuta una transacción que borra todos los datos de un cliente de la base de datos (incluido su carrito e historial de compra)
- **updPromo.sql**: script sql que crea una nueva columna promo en la tabla de customers.

Resultados y respuestas de los ejercicios.

1)NoSQL

Apartado a)

En este apartado creamos la nueva base de datos a través de la proporcionada en postgres. La consulta para obtener las películas que nos piden es la siguiente (se encuentra en el archivo database.py):

```
def db_topUSA():
    try:
        db_result = db.connect.execute("select distinct movietitle as title, year,\
array_to_string(array_agg(distinct genre), ', ') as genres,\
array_to_string(array_agg(distinct actorname), ', ') as actors\
from imdb_movies\
natural join imdb_moviecountries\
natural join imdb_moviegenres\
natural join imdb_actormovies\
natural join imdb_actors\
where country='USA' group by movietitle, year order by year desc, movietitle desc limit 800;")
        return list(db_result)
    except:
        print("Se va por except")
        return []

def db_getDirectors(movietitle):
    try:
        db_result = db.connect.execute("select directorname from imdb_directormovies natural join imdb_directors natural join imdb_movies where movietitle='"+movietitle+"'")
        return list(db_result)
    except:
        return []
```

En la primera consulta no recogemos los directores, ya que hay algunas películas que no tienen director y por tanto no mostraría esas mismas películas. Para solucionarlo, hemos creado otra consulta que mas adelante recoge los directores de cada película, la cual la pasamos por referencia.

Apartado b)

En este apartado se nos pide, con los datos anteriores, desarrollar el código para almacenar esos mismos datos en una base de datos en MongoDB llamada si1, que contenga una colección que se llame topUSA y con el código realizado, crear un script en Python llamado createMongoDBFromPostgreSQLDB.py

En la siguiente captura de pantalla se muestra el código desarrollado y su correcto funcionamiento:

```
import pymongo
import sys
import inspect
import os
current_dir = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())))
parent_dir = os.path.dirname(current_dir)
sys.path.insert(0, parent_dir)
import database
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["si1"]
if 'si1' in client.list_database_names():
    db.collection.drop()
    client.drop_database("si1")
else:
    print("La base de datos no estaba creada.")
collection = db["topUSA"]
movies = database.db_topUSA()

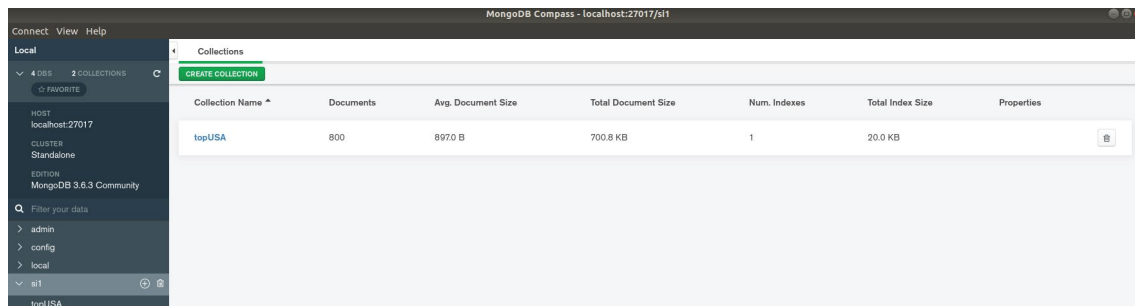
for item in movies:
    dict = {"title": item[0],
            "year": item[1],
            "genres": item[2],
            "directors": "",
            "actors": item[3]}
    directors=database.db_getDirectors(item[0])
    if directors:
        for x in directors:
            dict["directors"]=dict["directors"]+str(x)
    else:
        dict["directors"]="Sin un director especifico"
    x = collection.insert_one(dict)
```

```
def mongo_Life():
    try:
        query=db.topUSA.find({"title":{"$regex":"Life"}});
        return list(query)
    except:
        print("Error Life")
        return []

def mongo_Allen():
    try:
        query=db.topUSA.find({"directors":{"$regex":"Allen, Woody"},"year":{"$regex":"199"}});
        return list(query)
    except:
        print("Error Allen")
        return []

def mongo_ParsonsGalecki():
    try:
        query=db.topUSA.find({"$and":[{"actors":{"$regex":"Parsons"}},{"actors":{"$regex":"Galecki"}}]});
        return list(query)
    except:
        print("Error Parsons")
        return []
```

Podemos ver en MongoDB Compass que ha creado la base de datos y la colección con 800 documentos.



Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
topUSA	800	8970 B	700.8 KB	1	20.0 KB	

Apartado c)

En este apartado debemos crear un nuevo enlace en nuestra página web que conecte con la base de datos creada en MongoDB que se llame “Top USA”, que mostramos a continuación:



Sistema de venta online

Log Out | | Enviar | Carrito

Historial de usuario

Top USA

Catalogo

46 usuarios

'Breaker' Morant (1980)	'Crocodile' Dundee II (1988)	'Crocodile' Dundee II (1988)

Tu pagina web de confianza para comprar y alquilar DVDs - Web realizada por Luis Miguel y Adrian para SI

Apartado d)

En este apartado tenemos que crear una nueva página html a la que nos llevará cuando pinchemos en el enlace anterior, y que nos muestre las 3 tablas pedidas en el enunciado, como se muestra a continuación:

Sistema de venta online																																							
<div>Log Out</div> <div>Historial de usuario</div> <div>Top USA</div> <div>Catalogo</div> <div>100 usuarios</div>		<div>Carrito</div> <div>Tabla Life</div> <table><tr><th>TITLE</th><td>After Life (2005)</td><td>My Life So Far (1999)</td><td>Life (1999/I)</td><td>Life and Times of Hank Greenberg, The (1998)</td><td>Bug's Life, A (1998)</td><td>Life Less Ordinary, A (1997)</td><td>Life During Wartime (1997)</td></tr><tr><th>GENRES</th><td>Comedy, Short</td><td>Biography, Comedy, Drama</td><td>Comedy, Drama</td><td>Biography, Documentary, Sport</td><td>Adventure, Animation, Comedy, Family</td><td>Comedy, Crime, Drama, Fantasy, Romance</td><td>Comedy</td></tr><tr><th>YEAR</th><td>2005</td><td>1999</td><td>1999</td><td>1998</td><td>1998</td><td>1997</td><td>1997</td></tr><tr><td></td><td></td><td>Anderson, Ross (III); Baird, Daniel; Barlow, Eric; Bett, John (I); Boni, Lorenzo; Campbell, Victoria (I); Cochrane, Robyn; Ellis, Elaine M.; Fergie, Jennifer; Firth, Colin; Forrest, Stewart; Foulds, Jenny; Gleeson, Brendan; Harris, Rosemary (I); Hart, Andrea; Hunter, Moray;</td><td>Alexander, David (IV); Almy, Brooks; Anderson, Anthony (I); Anderson III, Haskell V.; Babatundé, Obba; Banks, Ernie Lee; Beatty, Ned; Benton, Shayne; Blunt, Augie; Boyles, Steve; Brooks, Hildy; Brown, Johnny; Brubaker, James D.; Burke, Keith; Call, Allyson; Carlson, Lisa Nicole; Cassavetes, Nick (I); Colston, Sydney 'Big Dawg'; Compton, O'Neal; DeMilo, Venus;</td><td>Allen, Harold (II); Auker, Elden; Baker, Del (II); Barahal, Dr. George; Berkow, Ira; Brenner, Rabbi Reeve; Bridges, Tommy; Briggs, Basil 'Mickey'; Briggs Jr., Walter O.; Briggs Sr., Walter O.; Chandler, Albert 'Happy'; Clifton, Herman 'Flea'; Cochrane, Mickey; Collins, Rip; Colman, Harriet; Coughlin, Father, Dean, Dizzy; Derringer, Paul; Dershowitz, Alan M.; DiMaggio, Joe; Eisenstat, Harry; Falls, Joe; Feller,</td><td>Alazraqui, Carlos; Angel, Jack (I); Bergen, Bob; Brown, Kimberly J.; Bumpass, Rodger; Burnh, Anthony; Darling, Jennifer; Davey, Rachel; Derynberg, Debi; Diller, Phyllis (I); Eiding, Paul; Evans, Jessica (I); Farmer, Bill (I); Foley, Dave (I); Garrett, Brad (I); Gifaldi, Sam; Hall, Brad (I); Harnell,</td><td>Brunson, Kitty; Byrd, Toni Lynn; Chaykin, Maury; Craven, Jayceen; Decker, Anne Cullimore; Diaz, Cameron; Dodds, K.K.; Gorman, Christopher; Gowdy, Chuck;</td><td>Arquette, David; Arquette, Lewis; Arquette, Richmond; Bilancio, Vinnie; Brisbin, David (I); Campbell, Colin (I); Capshaw, Kate;</td></tr></table>						TITLE	After Life (2005)	My Life So Far (1999)	Life (1999/I)	Life and Times of Hank Greenberg, The (1998)	Bug's Life, A (1998)	Life Less Ordinary, A (1997)	Life During Wartime (1997)	GENRES	Comedy, Short	Biography, Comedy, Drama	Comedy, Drama	Biography, Documentary, Sport	Adventure, Animation, Comedy, Family	Comedy, Crime, Drama, Fantasy, Romance	Comedy	YEAR	2005	1999	1999	1998	1998	1997	1997			Anderson, Ross (III); Baird, Daniel; Barlow, Eric; Bett, John (I); Boni, Lorenzo; Campbell, Victoria (I); Cochrane, Robyn; Ellis, Elaine M.; Fergie, Jennifer; Firth, Colin; Forrest, Stewart; Foulds, Jenny; Gleeson, Brendan; Harris, Rosemary (I); Hart, Andrea; Hunter, Moray;	Alexander, David (IV); Almy, Brooks; Anderson, Anthony (I); Anderson III, Haskell V.; Babatundé, Obba; Banks, Ernie Lee; Beatty, Ned; Benton, Shayne; Blunt, Augie; Boyles, Steve; Brooks, Hildy; Brown, Johnny; Brubaker, James D.; Burke, Keith; Call, Allyson; Carlson, Lisa Nicole; Cassavetes, Nick (I); Colston, Sydney 'Big Dawg'; Compton, O'Neal; DeMilo, Venus;	Allen, Harold (II); Auker, Elden; Baker, Del (II); Barahal, Dr. George; Berkow, Ira; Brenner, Rabbi Reeve; Bridges, Tommy; Briggs, Basil 'Mickey'; Briggs Jr., Walter O.; Briggs Sr., Walter O.; Chandler, Albert 'Happy'; Clifton, Herman 'Flea'; Cochrane, Mickey; Collins, Rip; Colman, Harriet; Coughlin, Father, Dean, Dizzy; Derringer, Paul; Dershowitz, Alan M.; DiMaggio, Joe; Eisenstat, Harry; Falls, Joe; Feller,	Alazraqui, Carlos; Angel, Jack (I); Bergen, Bob; Brown, Kimberly J.; Bumpass, Rodger; Burnh, Anthony; Darling, Jennifer; Davey, Rachel; Derynberg, Debi; Diller, Phyllis (I); Eiding, Paul; Evans, Jessica (I); Farmer, Bill (I); Foley, Dave (I); Garrett, Brad (I); Gifaldi, Sam; Hall, Brad (I); Harnell,	Brunson, Kitty; Byrd, Toni Lynn; Chaykin, Maury; Craven, Jayceen; Decker, Anne Cullimore; Diaz, Cameron; Dodds, K.K.; Gorman, Christopher; Gowdy, Chuck;	Arquette, David; Arquette, Lewis; Arquette, Richmond; Bilancio, Vinnie; Brisbin, David (I); Campbell, Colin (I); Capshaw, Kate;
TITLE	After Life (2005)	My Life So Far (1999)	Life (1999/I)	Life and Times of Hank Greenberg, The (1998)	Bug's Life, A (1998)	Life Less Ordinary, A (1997)	Life During Wartime (1997)																																
GENRES	Comedy, Short	Biography, Comedy, Drama	Comedy, Drama	Biography, Documentary, Sport	Adventure, Animation, Comedy, Family	Comedy, Crime, Drama, Fantasy, Romance	Comedy																																
YEAR	2005	1999	1999	1998	1998	1997	1997																																
		Anderson, Ross (III); Baird, Daniel; Barlow, Eric; Bett, John (I); Boni, Lorenzo; Campbell, Victoria (I); Cochrane, Robyn; Ellis, Elaine M.; Fergie, Jennifer; Firth, Colin; Forrest, Stewart; Foulds, Jenny; Gleeson, Brendan; Harris, Rosemary (I); Hart, Andrea; Hunter, Moray;	Alexander, David (IV); Almy, Brooks; Anderson, Anthony (I); Anderson III, Haskell V.; Babatundé, Obba; Banks, Ernie Lee; Beatty, Ned; Benton, Shayne; Blunt, Augie; Boyles, Steve; Brooks, Hildy; Brown, Johnny; Brubaker, James D.; Burke, Keith; Call, Allyson; Carlson, Lisa Nicole; Cassavetes, Nick (I); Colston, Sydney 'Big Dawg'; Compton, O'Neal; DeMilo, Venus;	Allen, Harold (II); Auker, Elden; Baker, Del (II); Barahal, Dr. George; Berkow, Ira; Brenner, Rabbi Reeve; Bridges, Tommy; Briggs, Basil 'Mickey'; Briggs Jr., Walter O.; Briggs Sr., Walter O.; Chandler, Albert 'Happy'; Clifton, Herman 'Flea'; Cochrane, Mickey; Collins, Rip; Colman, Harriet; Coughlin, Father, Dean, Dizzy; Derringer, Paul; Dershowitz, Alan M.; DiMaggio, Joe; Eisenstat, Harry; Falls, Joe; Feller,	Alazraqui, Carlos; Angel, Jack (I); Bergen, Bob; Brown, Kimberly J.; Bumpass, Rodger; Burnh, Anthony; Darling, Jennifer; Davey, Rachel; Derynberg, Debi; Diller, Phyllis (I); Eiding, Paul; Evans, Jessica (I); Farmer, Bill (I); Foley, Dave (I); Garrett, Brad (I); Gifaldi, Sam; Hall, Brad (I); Harnell,	Brunson, Kitty; Byrd, Toni Lynn; Chaykin, Maury; Craven, Jayceen; Decker, Anne Cullimore; Diaz, Cameron; Dodds, K.K.; Gorman, Christopher; Gowdy, Chuck;	Arquette, David; Arquette, Lewis; Arquette, Richmond; Bilancio, Vinnie; Brisbin, David (I); Campbell, Colin (I); Capshaw, Kate;																																
Tu pagina web de confianza para comprar y alquilar DVDs - Web realizada por Luis Miguel y Adrian para SI																																							

Tu pagina web de confianza para comprar y alquilar DVDs - Web realizada por Luis Miguel y Adrian para SI

Sistema de venta online

Log Out

Historial de usuario

Top USA

Catalogo

67 usuarios

TITLE

GENRES

YEAR

Sweet and Loudown (1999)

Comedy, Drama, Music

1999

ACTORS

Addy, William; Allen, Woody; Ambrose, Joe (I); Bargonetti, Steve; Bauer, Chris (I); Baynes, Carlo; Bolan, Michael; Brown, Benjamin Franklin; Bryant, Al; Ciccasi, Michael; Cossin, Stanley; Danerger, Anna; Darrow, Tony; Davis, Eddy; Duffley, Catherine; Duncan, Ben (I); Edelson, Kenneth; Ezz, Lutz; Erskine, Drummond; Francis, Alan (III); Garrett, Bird (I); Garvey, Ray; Grayson, Clark; Giles (I); Brooks, Giordano; Vines; Goehner, Fred; Gustafson, Vincent; Hamill, Kati; Henoff, Nat; Johnson, Brian; Madson, Jones; Ron, Alex; Kellin, Douglas; Kemp, Emma; LaPaglia, Anna; Leung, Peter (II); Lewkowicz, Chuck; Markinson, Mark; McGretter, Trish; McGrath, Douglas; McQuinn, Earl P.; McLaughlin, John Patrick; McLaughlin, Marcus; Mercutio, William; Mingalone, Dick; Mott, Gretchen; Munday, Dick; Moran, Dan (I); Morton, Samantha; Moore, Rick; Mowery, John (I); O'Hare, Denis; Okrent, David; Ofaroke, Dick (I); Ottavio, Maria Overby; Kellie; Parrish, Paul; Padalinski, Lella Penn; Stein (I); Placksin, Sally; Pope, Ralph; Price, Molly; Reed, Randy; Alan (I); Richardson, Jerome (I); Rigano, Joseph (I); Sacchetti Jr., Alfred; Saxton, Carolyn; Schmitt, August; Siez, Adam; Solari; Lough, Spoons, Mr; Sprague, Michael (I); Stein, Dennis; Stout, Mary; Thurman, Uma; Tolan, John (I); Ubrankin, James; Vernoff, Kelli; Waters, John (II); Woss, Frank; Wetzelsh, Simon; Wilkins, Ted; Williams, James (XII); Woods, Carol

DIRECTORS

(Allen, Woody.)

Stanley

(Boyle, Danny.)

(Dusky, E.)

Sin un director específico

Tabla Woody Allen

TITLE

GENRES

YEAR

Sweet and Loudown (1999)

Comedy

1998

ACTORS

Adams, Polly (II); Addison, Bernard (I); Addy, William; Affle, Ron; Alexi-Male, Andrea; Adams, Casper; Aflack, Neal; Aronoff, Ethan; Azaria, Hank; Bagley, Barker, Becky; Amy; Baker, Dylan (I); Baxter, Tommie; Becker, Gerry; Beck, Marvin; Barbour, Jennifer; Bern, Berta; Messner, Arthur; Bible, Vladimir (I); Billingham, Dar; Bonausage, Frank; Boyden, Peter; Brannagh, Kenneth; Bird, Sarah; Burke, Marylouise; Burton, Kate (I); Buttafuoco, John; Buttafuoco, John Cabon; Angel; Carter, John (IV); Castelloni, Peter; Cavanaugh, New York; Cox; Civalano, Robert; Cohen, Ray; Colebach, Victor; Cogan, Tracy; Costelloe, John; Cowling, Gary; Crecro, Michael; Cucchioli, Robert; Cuervo, Alma; Dalosie, Sam; D'Abraville, Paul; Dark, Peter; Darrow, Tony; Davis, Judy; Dell'Orefice, Carmen; De Lucia, Amy; DePrist, Tere; DeVega, Sam; DeCaprio, Leonardo; Dossanain, John; Doyle, Kathleen; Duffy, Karen (I); Edelson, Kenneth; Ezz, Lutz; Eisenberg, Ned (I); Jay, Melinda; Erskine, Howard; Fitzgibbon, Ramsey; Fitzgerald, Douglas; Ford, Cebert; Fowler, Monique; Friedman, Bruce Jay; Gerber, Bill (I); Gestone, Anthony; Getz, Elin; Gossilid, Bruce; Glazov, Irene; Gary, Sam (I); Gregory, Andre; Grenier, Adrian; Griffith, Melanie; Haddon, David; Halston, Jale; Harvor, David; Harrell, Robert; Harris, Wood; Hawk, Kirk; Iacona, Richard; Idd, Markanthony; Jackson, Rebecca; Jannay, Allison; Janssen, Fankie; Johnson, Timothy; Johnson, Woody (IV); Jong, Erica; Jordan, Randy; Kainer, David (II); Kell, Michael; Knobs, Sander; Kiehlner, Tom; Kirk, Kritt; Kulick, Kyle; Lerner, Michael (I); Levy, Nicole (I); Lipari, Renee; Long, Jodi; Lucas, Autunum; Mann, Heather; Marnetown, Joe; Margulies, David; Mark, Kevin; Marlow, James; Mason, Anthony (II); Mawe, Richard; Mazzola, Jeff; McCarthy, Patrick (I); McCombsine, Brian; McCormack, Brian; McGretter, Trish; McGrath, Douglas; McKeown, Dennis; McNamara, Maureen (I); McRobbie, Peter; Meer; Altan; Mellor; Steve; Merano, Elizabeth; Messing, David; Milman, Gabriel; Mingalone, Dick; Mizrahi, Issa; (I); Mott, Gretchen; Moody, Jim (I); Moon, Michael (I); Moran, Dan (I); Morris, A. L.; Mottola, George; Mowat, Rick; Natvo, Laina; Neustadt, Ted; Newirth, Bette; O'Connor, Murphy; Oh, Jina; Ott; Krich; Palladino, Aleks; Pantavio, Irina; Pellegrino, Frank (II); Perky, Stan; Pine, Larry; Poccia, Eric; Poppella, Mike; Pope, Ralph; Prince, Datusy; Prizdale, Francisco; Rios, Bill; Randolph, David; Rappaport, David; Reynolds, Eve; Richards, Jay (I); Rockwell, Sam; Rogers, Ingrid; Rolf, Frederick; Rose, Skip; Ross, James; Ross, Yolanda; Royce; Winnon; Saint, Louis Augustin; Craystie; Sallvach, Eve; Schneiburger, Peter; Schwartzberg, Antonette; Scallin, Kevin; Selby, Marian; Shenk, Les; Siez, Adam; Simmons, J.K.; Siriso, Tony; Sloan, Tina; Somerschilder, Ira; Sonenberg, Maurice; Steinman, Robert; Sullivan, G.D.; Sweeney, Matthew; Tedeoos, Tony (I); Theron, Charlize; Torres, Robert (I); Trump, Donald; Tullio, Joseph; Turturo, Aldo; Ulmschneider, Craig; Valencia, Carlos (II); Vandelooer, Mark; Van Der Wal, Frederique; Weston, Celea; Williams, Giti; Williams, Roshumba; Wright, Jeffrey (I); Wright, Mary; Catherine; Colloff; Diana C.

Tu pagina web de confianza para comprar y alquilar DVDs - Web realizada por Luis Miguel y Adrian para SI

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	
	ACTORS	
	93 usuarios	
	DIRECTORS	
	TITLE	
	GENRES	
YEAR		
ACTORS		
DIRECTORS		

Sistema de venta online		
	YEAR	
Log Out Historial de usuario Top USA Catalogo	1999	

2) Optimización

Apartado e)

En este apartado debemos estudiar el impacto de un índice mediante la creación de una consulta que se llama clientesDistintos.sql. Para realizar esta consulta se nos pedía que usásemos la sentencia EXPLAIN, que mejorásemos el rendimiento de la consulta, probar distintos índices y discutir los resultados. Después del proceso de discusión, los resultados son:

- **Crear una consulta clientesDistintos.sql**

```
select count(distinct customerid) as Suma
  from orders
 where date_part('month',orderdate)= 04 and
        date_part('year',orderdate) = 2015 and totalamount > 100;
```

- **Sentencia EXPLAIN**

```
QUERY PLAN
-----
Aggregate  (cost=5627.93..5627.94 rows=1 width=8)
->  Gather  (cost=1000.00..5627.92 rows=2 width=4)
      Workers Planned: 1
        -> Parallel Seq Scan on orders  (cost=0.00..4627.72 rows=1 width=4)
              Filter: ((totalamount > '100'::numeric) AND (date_part('month'::text,
(orderdate)::timestamp without time zone) = '4'::double precision) AND (date_part('
year'::text, (orderdate)::timestamp without time zone) = '2015'::double precision))
(5 filas)
```

Al hacer el explain, observamos que el coste de la consulta es de 5627.

- **Identificar un índice que mejore la consulta y crearlo**

```
sql=# create index totalamount_index on orders (totalamount);
CREATE INDEX
```

Hemos concluido que el índice que mejora el rendimiento es el que se realiza sobre totalamount, ya que al probar con orderdate el coste no variaba.

- **Estudiar el nuevo plan de ejecución y compararlo con el anterior.**

```
QUERY PLAN
-----
Aggregate  (cost=4480.32..4480.33 rows=1 width=8)
->  Bitmap Heap Scan on orders  (cost=1126.90..4480.32 rows=2 width=4)
      Recheck Cond: (totalamount > '100'::numeric)
      Filter: ((date_part('month'::text, (orderdate)::timestamp without time zone) = '4'::double precision) AND (date_part('year'::text, (orderdate)::timestamp without time zone) = '2015'::double precision))
->  Bitmap Index Scan on totalamount_index  (cost=0.00..1126.90 rows=60597 width=0)
      Index Cond: (totalamount > '100'::numeric)
(6 filas)
```

Como podemos observar, el coste de la consulta ha mejorado, pasando a valer 4480

- **Modificar el script clientesDistintos.sql**

```
explain
select count(distinct customerid) as Suma
  from orders
 where date_part('month',orderdate)= 04 and
        date_part('year',orderdate) = 2015 and totalamount > 100;
```

Apartado f)

Este apartado es únicamente de discusión y estudio, se trata principalmente de estudiar el impacto en la ejecución de unas sentencias sql que se nos incluyen en el guion de la práctica.

- Primera sentencia:

```
si1=# explain select customerid
si1=# from customers
si1=# where customerid not in (
si1(# select customerid
si1(# from orders
si1(# where status='Paid'
si1(# );

               QUERY PLAN
-----
Seq Scan on customers (cost=3961.65..4490.81 rows=7046 width=4)
  Filter: (NOT (hashed SubPlan 1))
    SubPlan 1
      -> Seq Scan on orders (cost=0.00..3959.38 rows=909 width=4)
          Filter: ((status)::text = 'Paid'::text)
(5 filas)
```

- Segunda sentencia:

```
si1=# explain select customerid
si1=# from (
si1(# select customerid
si1(# from customers
si1(# union all
si1(# select customerid
si1(# from orders
si1(# where status='Paid'
si1(# ) as A
si1=# group by customerid
si1=# having count(*) =1;

               QUERY PLAN
-----
HashAggregate (cost=4537.41..4539.41 rows=200 width=4)
  Group Key: customers.customerid
  Filter: (count(*) = 1)
    -> Append (cost=0.00..4462.40 rows=15002 width=4)
        -> Seq Scan on customers (cost=0.00..493.93 rows=14093 width=4)
        -> Seq Scan on orders (cost=0.00..3959.38 rows=909 width=4)
            Filter: ((status)::text = 'Paid'::text)
(7 filas)
```

- Tercera sentencia:

```
si1=# explain select customerid
si1=# from customers
si1=# except
si1=# select customerid
si1=# from orders
si1=# where status='Paid';

               QUERY PLAN
-----
HashSetOp Except (cost=0.00..4640.83 rows=14093 width=8)
  -> Append (cost=0.00..4603.32 rows=15002 width=8)
      -> Subquery Scan on "*SELECT* 1" (cost=0.00..634.86 rows=14093 width=8)
          -> Seq Scan on customers (cost=0.00..493.93 rows=14093 width=4)
      -> Subquery Scan on "*SELECT* 2" (cost=0.00..3968.47 rows=909 width=8)
          -> Seq Scan on orders (cost=0.00..3959.38 rows=909 width=4)
              Filter: ((status)::text = 'Paid'::text)
(7 filas)
```


Después de discutirlo, hemos decidimos crear un índice en orders(status) de la siguiente manera:

```
si1=# create index status_orders on orders(status);  
CREATE INDEX
```

Y volvemos a lanzar las consultas de nuevo después de crear el índice.

- Primera consulta:

```
si1=# explain select customerid  
si1-# from customers  
si1-# where customerid not in (  
si1-# select customerid  
si1-# from orders  
si1-# where status='Paid'  
si1-# );  
  
QUERY PLAN  
-----  
Seq Scan on customers (cost=1498.79..2027.96 rows=7046 width=4)  
  Filter: (NOT (hashed SubPlan 1))  
  SubPlan 1  
    -> Bitmap Heap Scan on orders (cost=19.46..1496.52 rows=909 width=4)  
        Recheck Cond: ((status)::text = 'Paid'::text)  
        -> Bitmap Index Scan on status_orders (cost=0.00..19.24 rows=909 width=0)  
            Index Cond: ((status)::text = 'Paid'::text)  
(7 filas)
```

- Segunda consulta:

```
si1=# explain select customerid  
si1-# from (  
si1-# select customerid  
si1-# from customers  
si1-# union all  
si1-# select customerid  
si1-# from orders  
si1-# where status='Paid'  
si1-# ) as A  
si1-# group by customerid  
si1-# having count(*) =1;  
  
QUERY PLAN  
-----  
HashAggregate (cost=2074.55..2076.55 rows=200 width=4)  
  Group Key: customers.customerid  
  Filter: (count(*) = 1)  
    -> Append (cost=0.00..1999.54 rows=15002 width=4)  
        -> Seq Scan on customers (cost=0.00..493.93 rows=14093 width=4)  
        -> Bitmap Heap Scan on orders (cost=19.46..1496.52 rows=909 width=4)  
            Recheck Cond: ((status)::text = 'Paid'::text)  
            -> Bitmap Index Scan on status_orders (cost=0.00..19.24 rows=909 width=0)  
                Index Cond: ((status)::text = 'Paid'::text)  
(9 filas)
```

- Tercera consulta:

```
si1=# explain select customerid  
si1-# from customers  
si1-# except  
si1-# select customerid  
si1-# from orders  
si1-# where status='Paid';  
  
QUERY PLAN  
-----  
HashSetOp Except (cost=0.00..2177.98 rows=14093 width=8)  
  -> Append (cost=0.00..2140.47 rows=15002 width=8)  
      -> Subquery Scan on ""SELECT* 1" (cost=0.00..634.86 rows=14093 width=8)  
          -> Seq Scan on customers (cost=0.00..493.93 rows=14093 width=4)  
      -> Subquery Scan on ""SELECT* 2" (cost=19.46..1505.61 rows=909 width=8)  
          -> Bitmap Heap Scan on orders (cost=19.46..1496.52 rows=909 width=4)  
              Recheck Cond: ((status)::text = 'Paid'::text)  
              -> Bitmap Index Scan on status_orders (cost=0.00..19.24 rows=909 width=0)  
                  Index Cond: ((status)::text = 'Paid'::text)  
(9 filas)
```


Como podemos observar, hemos logrado una mejoría en los tiempos a la hora de añadir el índice anterior.

- ¿Qué consulta devuelve algún resultado nada más comenzar su ejecución?
La última consulta, ya que su coste al comenzar es prácticamente de cero.
- ¿Qué consulta se puede beneficiar de la ejecución en paralelo?
En la segunda consulta, ya que las subconsultas que contiene se pueden paralelizar.

Apartado g)

Este apartado se basa en estudio del impacto de la generación de estadísticas, discusión y creación de un script countStatus.sql. Para comenzar, debemos estudiar y discutir las sentencias que se nos incluyen en el guion de la práctica (antes de ejecutar las sentencias hemos reiniciado la base de datos):

- Primera consulta:

```
si1=# explain select count(*)
si1=# from orders
si1=# where status is null;
               QUERY PLAN
-----
Aggregate  (cost=3507.17..3507.18 rows=1 width=8)
-> Seq Scan on orders  (cost=0.00..3504.90 rows=909 width=0)
    Filter: (status IS NULL)
(3 filas)
```

- Segunda consulta:

```
si1=# explain select count(*)
si1=# from orders
si1=# where status = 'Shipped';
               QUERY PLAN
-----
Aggregate  (cost=3961.65..3961.66 rows=1 width=8)
-> Seq Scan on orders  (cost=0.00..3959.38 rows=909 width=0)
    Filter: ((status)::text = 'Shipped'::text)
(3 filas)
```

Tras crear el índice en orders(status):

```
si1=# create index status_orders on orders(status);
CREATE INDEX
```

- Primera consulta:

```
si1=# explain select count(*)
si1=# from orders
si1=# where status is null;
               QUERY PLAN
-----
Aggregate  (cost=1496.52..1496.53 rows=1 width=8)
-> Bitmap Heap Scan on orders  (cost=19.46..1494.25 rows=909 width=0)
    Recheck Cond: (status IS NULL)
-> Bitmap Index Scan on status_orders  (cost=0.00..19.24 rows=909 width=0)
    Index Cond: (status IS NULL)
(5 filas)
```

- Segunda consulta:

```

si1=# explain select count(*)
si1=# from orders
si1=# where status = 'Shipped';
                                QUERY PLAN
-----
Aggregate  (cost=1498.79..1498.80 rows=1 width=8)
-> Bitmap Heap Scan on orders  (cost=19.46..1496.52 rows=909 width=0)
    Recheck Cond: ((status)::text = 'Shipped'::text)
-> Bitmap Index Scan on status_orders  (cost=0.00..19.24 rows=909 width=0)
    Index Cond: ((status)::text = 'Shipped'::text)
(5 filas)

```

Generamos las estadísticas:

```

si1=# analyze;
ANALYZE

```

ANALYZE sin ningún parámetro analiza todas las tablas de la base de datos actual. Ahora, ejecutamos las dos consultas anteriores de nuevo más las dos siguientes que se nos indican en el guion.

- Primera consulta:

```

si1=# explain select count(*)
si1=# from orders
si1=# where status is null;
                                QUERY PLAN
-----
Aggregate  (cost=7.26..7.27 rows=1 width=8)
-> Index Only Scan using status_orders  (cost=0.42..7.26 rows=1 width=0)
    Index Cond: (status IS NULL)
(3 filas)

```

- Segunda consulta:

```

si1=# explain select count(*)
si1=# from orders
si1=# where status = 'Shipped';
                                QUERY PLAN
-----
Finalize Aggregate  (cost=4210.32..4210.33 rows=1 width=8)
-> Gather  (cost=4210.20..4210.31 rows=1 width=8)
    Workers Planned: 1
-> Partial Aggregate  (cost=3210.20..3210.21 rows=1 width=8)
-> Parallel Seq Scan on orders  (cost=0.00..3023.69 rows=74605 width=0)
    Filter: ((status)::text = 'Shipped'::text)
(6 filas)

```

- Tercera consulta:

```

si1=# explain select count(*)
si1=# from orders
si1=# where status = 'Paid';
                                QUERY PLAN
-----
Aggregate  (cost=2310.37..2310.38 rows=1 width=8)
-> Bitmap Heap Scan on orders  (cost=355.05..2265.65 rows=17888 width=0)
    Recheck Cond: ((status)::text = 'Paid'::text)
-> Bitmap Index Scan on status_orders  (cost=0.00..350.58 rows=17888 width=0)
    Index Cond: ((status)::text = 'Paid'::text)
(5 filas)

```

- Cuarta consulta:

```

s11=# explain select count(*)
s11=# from orders
s11=# where status ='Processed';
                                QUERY PLAN
-----
Aggregate  (cost=2974.83..2974.84 rows=1 width=8)
-> Bitmap Heap Scan on orders  (cost=731.74..2882.15 rows=37073 width=0)
    Recheck Cond: ((status)::text = 'Processed'::text)
-> Bitmap Index Scan on status_orders  (cost=0.00..722.47 rows=37073 width=0)
    Index Cond: ((status)::text = 'Processed'::text)
(5 filas)

```

Conclusiones: El generador de estadísticas “ANALYZE” sirve para analizar los datos de la base de datos o tabla que indiquemos para optimizar las búsquedas. La diferencia entre las consultas antes de hacer el analyze y después viene dada porque tras analizar los datos, cuando hay un status que ocupa un gran porcentaje de las filas, es ineficiente tenerlo en el índice, ya que es más costoso mantenerlo que buscar en todas las filas.

3) Transacciones

Apartado h)

Como ejemplo de transacciones en la base de datos, queremos borrar los datos de usuario con ID 15. Para empezar, provocaremos un fallo de integridad que anule la transacción y haga rollback.

Ejemplo de Transacción con Flask SQLAlchemy

Customer ID:

☒ Transacción vía sentencias SQL
☐ Transacción vía funciones SQLAlchemy

☐ Ejecutar commit intermedio
☒ Provocar error de integridad

Duerme segundos (para forzar deadlock).

Trazas

Al darle a enviar, nos saldrán estas trazas:

Trazas

1. Se ejecuta a través de SQL
2. BEGIN
3. Se borrarán los datos de los pedidos del cliente: 15
4. Error en la transacción. Haciendo rollback.
5. Cerramos la conexión con la base de datos.

Podemos comprobar que los datos de los pedidos efectivamente no se han borrado a través de phpPGAdmin:

PostgreSQL 10.13 (Ubuntu 10.13-1.pgdg18.04+1) corriendo en localhost:5432 -- Usted ha iniciado sesión con el usuario "alumnodb"

phpPgAdmin: PostgreSQL: si1?

Esquemas? SQL? Buscar Variables? Procesos?

Ingrese la sentencia de SQL para ejecutar:

SQL

```
select count(*) from orderdetail where orderid in (select orderid from orders where customerid=15);
```

o subir un script SQL: Ningún archivo seleccionado

☒ Pagar resultados

Y como podemos observar, siguen saliendo los pedidos:

PostgreSQL 10.13 (Ubuntu 10.13-1.pgdg18.04+1) corriendo en localhost:5432 -- Usted ha iniciado sesión con el usuario "alumnodb"

phpPgAdmin: PostgreSQL: si1?

Resultado de la consulta

count
132

1 fila(s)

[Editar SQL](#) | [Expandir](#) | [Bajar](#) | [Refrescar](#)

Ahora vamos a probar a ejecutarlo con un fallo, pero haciendo commit intermedio:

Ejemplo de Transacción con Flask SQLAlchemy

Customer ID:

☒ Transacción vía sentencias SQL
☐ Transacción vía funciones SQLAlchemy

☐ Ejecutar commit intermedio

☒ Provocar error de integridad

Duerme segundos (para forzar deadlock).

Trazas

1. Se ejecuta a través de SQL
2. BEGIN
3. Se borrarán los datos de los pedidos del cliente: 15
4. Commit intermedio. Se inicia otra transacción
5. Error en la transacción. Haciendo rollback.
6. Cerramos la conexión con la base de datos.

Ahora comprobamos si se han borrado los pedidos en la base de datos con la sentencia anterior:

PostgreSQL 10.13 (Ubuntu 10.13-1.pgdg18.04+1) corriendo en localhost:5432 -- Usted ha iniciado sesión con el usuario "alumno_db"	
phpPgAdmin:	PostgreSQL?: si?
Resultado de la consulta	
count	0
1 fila(s)	
Editar SQL Expandir Bajar Refrescar	

Al hacer el commit intermedio, el rollback vuelve al momento después de borrar los detalles de los pedidos.

Vamos a probar a borrar otro usuario con un id distinto y sin provocar ningún error:

Ejemplo de Transacción con Flask SQLAlchemy

Customer ID:

☒ Transacción vía sentencias SQL
☐ Transacción vía funciones SQLAlchemy

☐ Ejecutar commit intermedio
☐ Provocar error de integridad

Duerme segundos (para forzar deadlock).

Trazas

1. Se ejecuta a través de SQL
2. BEGIN
3. Se borrarán los datos de los pedidos del cliente: 7
4. Se borrarán los pedidos del cliente: 7
5. Se borrará el cliente
6. Transacción finalizada correctamente.
7. Cerramos la conexión con la base de datos.

Como podemos observar, la transacción se ha realizado correctamente.

Si intentamos borrar de nuevo el mismo cliente, hará un rollback, ya que ya no existe.

Ejemplo de Transacción con Flask SQLAlchemy

Customer ID:

☒ Transacción vía sentencias SQL
☐ Transacción vía funciones SQLAlchemy

☐ Ejecutar commit intermedio
☒ Provocar error de integridad

Duerme segundos (para forzar deadlock).

Trazas

1. Se ejecuta a través de SQL
2. BEGIN
3. Error en la transacción. Haciendo rollback.
4. Cerramos la conexión con la base de datos.

(El funcionamiento con SQLAlchemy es el mismo que con SQL)

Apartado i)

Este es el script de creación del trigger desarrollado:

```
--Apartado b) creacion de la nueva columna promo
alter table customers add promo integer;
--Apartado c) creacion del trigger
CREATE OR REPLACE FUNCTION promo() returns TRIGGER
language plpgsql
as $$
DECLARE
BEGIN
update orders set totalamount = totalamount- totalamount * promo/100
  from customers
  inner join orders on customer.customerid=orders.customerid
  where customerid= NEW.customerid and status = null;
  return new;
  --Apartado d) modificar el trigger para que haga un sleep con pg_sleep
  perform pg_sleep(15);
END;
$$;

create trigger updPromo after update or insert on customers
for each row execute procedure promo();
```

Como podemos observar, se crea correctamente:

```
si1=# \i app/sql/updPromo.sql
ALTER TABLE
CREATE FUNCTION
CREATE TRIGGER
```

- **Explicación del deadlocks y como se pueden evitar:**
El deadlock se produce al realizar una transacción de borrado, ya que intenta borrar los datos de la tabla customers, pero esta tabla está bloqueada por el trigger que hemos creado en updPromo.sql, ya que realiza una actualización.
Podemos evitar los deadlocks reproduciendo el tiempo de ejecución de las transacciones y optimizarlas con índices.