--HW14-1  
-- CREATE TABLE employee  
-- (  
-- id bigserial primary key ,  
-- name varchar(30),  
-- department varchar(30),  
-- salary real  
-- )  
  
-- SELECT \* FROM employee  
  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Mohsen', 'RealMadrid', 9000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Reza', 'Inter', 5000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Hasan', 'Barcelona', 6000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Hossein', 'AC Milan', 7000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Ali', 'Benfica', 8000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Mahdi', 'Porto', 8300.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Sasan', 'Pirozi', 530.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Jalal', 'Esteghlal', 540.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Ali', 'Man City', 8900.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Sajad', 'Man United', 4600.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Taghi', 'Sepahan', 910.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Naghi', 'golgohar Sirjan', 200.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Mohammad', 'Juventus', 3000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Ali', 'Zob Ahan', 1000.0);  
-- INSERT INTO employee(name, department, salary)  
-- VALUES ('Ali', 'Bayern', 4400.0);  
  
-- --نمایش کارمندانی که اسمشون علی است  
-- SELECT \* FROM employee  
-- WHERE name = 'Ali'  
  
-- -- نمایش حقوق کارمندانی که زیر 1000.0 دریافتی دارن  
-- SELECT \* FROM employee  
-- WHERE salary < 1000.0  
  
-- --نمایش میان گین حقوق کارمندان هر بخش  
-- SELECT AVG(salary)  
-- FROM employee  
-- WHERE department = 'Esteghlal';  
------------------------------------------------------------------------------------------------------------------------  
--HW14-2

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

--HW14-2  
-- CREATE TABLE student  
-- (  
-- id bigserial primary key ,  
-- name varchar(30),  
-- department varchar(30),  
-- age int4  
-- )  
  
-- INSERT INTO student(name, department, age)  
-- VALUES('Ali','Math',23);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Hasan','physics',24);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Ali','Computer',21);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Reza','Math',20);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Zahra','Computer',19);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Sasan','physics',29);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Ali','Naft',18);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Sina','Gaz',27);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Qasem','Polymer',26);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Kazem','History',25);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Sara','Math',22);  
-- INSERT INTO student(name, department, age)  
-- VALUES('Sajad','History',17);  
-- INSERT INTO student(name, department, age)  
-- VALUES('AliReza','Math',28);  
  
-- --نام همه دانشجویان به ترت یب حروف الفبا  
-- SELECT \* FROM student  
-- ORDER BY name;  
  
-- --نام سه دانشجوی اول به ترتیب حروف الفبا  
-- SELECT \* FROM student  
-- WHERE id = 1 or id = 2 or id = 3  
-- ORDER BY name  
  
-- --میانگین سن دانشجویان  
-- SELECT AVG(age)  
-- FROM student  
  
-- --تعداد دانشجویان بالای 2۰ سال  
-- SELECT count(age) FROM student  
-- WHERE age > 20;  
  
-- --نمایش دانشجویانی که اسمشون علی است  
-- SELECT count(name) FROM student  
-- WHERE name like '%Ali%'  
  
-- --تعداد دانشجویان دپارتمان فیزیک  
-- SELECT count(department) FROM student  
-- WHERE department = 'physics'  
  
-- --تعداد دانشجویان هر دپارتمان  
-- SELECT count(name) FROM student  
-- WHERE department = 'Gaz';  
-- SELECT count(name) FROM student  
-- WHERE department = 'Naft';  
-- SELECT count(name) FROM student  
-- WHERE department = 'History';  
-- SELECT count(name) FROM student  
-- WHERE department = 'Math';  
-- SELECT count(name) FROM student  
-- WHERE department = 'physics';

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

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

-- --HW14-3-Q1  
-- SELECT \* FROM albums  
  
-- --Q2  
-- SELECT \* FROM customers  
-- ORDER BY LastName  
  
-- --Q3  
-- SELECT \* FROM tracks  
-- ORDER BY Bytes desc  
-- limit 5  
  
-- --Q4  
-- SELECT count(TrackId) FROM tracks  
  
-- --Q5  
-- SELECT \* FROM tracks  
-- WHERE Milliseconds > 500000  
  
-- --Q6  
-- SELECT \* FROM tracks  
-- ORDER BY Milliseconds  
  
-- --Q7  
-- SELECT \* FROM customers  
-- WHERE City = 'Prague'  
  
-- --Q8  
-- SELECT \* FROM customers  
-- WHERE Company not null  
  
-- --Q9  
-- SELECT \* FROM customers  
-- WHERE City <> 'India'  
  
-- --Q10  
-- SELECT \*,count(artists.Name) as countArtistTraks FROM artists  
-- inner join albums on artists.ArtistId = albums.ArtistId  
-- GROUP BY artists.Name  
-- ORDER BY countArtistTraks desc  
-- limit 1  
  
-- --Q11  
-- SELECT \* FROM customers  
-- WHERE Email LIKE '%Gmail%'  
  
--Q12  
-- SELECT \* FROM tracks  
-- WHERE UnitPrice = 0.99 or UnitPrice = 1.99  
  
-- --Q13  
-- SELECT \* FROM tracks  
-- WHERE AlbumId = 5  
  
-- --Q14  
-- SELECT \* FROM tracks  
-- inner join genres on tracks.GenreId = genres.GenreId  
-- WHERE genres.Name = 'Rock'  
  
-- --Q15  
-- SELECT \* FROM tracks  
-- inner join media\_types mt on tracks.MediaTypeId = mt.MediaTypeId  
-- inner join genres gen on tracks.GenreId = gen.GenreId  
-- WHERE gen.Name = 'Jazz' and mt.Name = 'AAC audio file'  
  
-- --Q16  
-- SELECT \* FROM playlist\_track  
-- inner join playlists on playlist\_track.PlaylistId = playlists.PlaylistId  
-- WHERE Name = '90’s Music'  
  
--Q17  
-- SELECT gen.Name,count(gen.Name) as numberOfFavoriteGen FROM tracks  
-- inner join genres gen on tracks.GenreId = gen.GenreId  
-- inner join invoice\_items invItem on tracks.TrackId = invItem.TrackId  
-- inner join invoices inv on invItem.InvoiceId = inv.InvoiceId  
-- inner join customers cust on inv.CustomerId = cust.CustomerId  
-- GROUP BY gen.Name  
-- ORDER BY numberOfFavoriteGen desc  
-- limit 1  
  
-- --Q18  
-- SELECT gen.Name,count(gen.Name) FROM tracks  
-- inner join genres gen on tracks.GenreId = gen.GenreId  
-- GROUP BY gen.Name  
  
-- --Q19  
-- SELECT Name,count(invItem.Quantity) as countSell FROM tracks  
-- inner join invoice\_items invItem on tracks.TrackId = invItem.TrackId  
-- GROUP BY invItem.InvoiceId  
-- ORDER BY countSell  
  
-- --Q20  
-- SELECT Name,count(invItem.Quantity) as countSell FROM tracks  
-- inner join invoice\_items invItem on tracks.TrackId = invItem.TrackId  
-- GROUP BY invItem.InvoiceId  
-- ORDER BY countSell desc  
-- limit 10  
  
-- --Q21  
-- SELECT artst.Name,count(invItem.Quantity) \* tracks.UnitPrice as rich FROM tracks  
-- inner join invoice\_items invItem on tracks.TrackId = invItem.TrackId  
-- inner join albums al on tracks.AlbumId = al.AlbumId  
-- inner join artists artst on al.ArtistId = artst.ArtistId  
-- GROUP BY artst.Name  
-- ORDER BY rich  
-- limit 1