DROP TABLE IF EXISTS [dbo].[Employees]

GO

CREATE TABLE [dbo].[Employees](

[Id] [int] NULL,

[First\_Name] [varchar](50) NULL,

[Last\_Name] [varchar](50) NULL,

[Department] [varchar](50) NULL,

[Salary] [numeric](18, 0) NULL,

[Manager\_Id] [smallint] NULL

) ON [PRIMARY]

GO

INSERT [dbo].[Employees] ([Id], [First\_Name], [Last\_Name], [Department], [Salary], [Manager\_Id]) VALUES

(1, N'Angelika', N'Voules', N'Marketing', CAST(5294 AS Numeric(18, 0)), 2)

,(2, N'Rozelle', N'Swynley', N'Marketing', CAST(8295 AS Numeric(18, 0)), 18)

,(3, N'Warren', N'Willey', N'Engineering', CAST(9127 AS Numeric(18, 0)), 19)

,(4, N'Lynelle', N'Whiten', N'Management Board', CAST(10716 AS Numeric(18, 0)), NULL)

,(5, N'Consolata', N'Roman', N'Legal', CAST(8456 AS Numeric(18, 0)), 4)

,(6, N'Hoebart', N'Baldock', N'Research and Development', CAST(4817 AS Numeric(18, 0)), 20)

,(7, N'Starlene', N'Watkiss', N'Accounting', CAST(6541 AS Numeric(18, 0)), 4)

,(8, N'Barde', N'Ribbens', N'Marketing', CAST(4853 AS Numeric(18, 0)), 2)

,(9, N'Lorne', N'Philipsen', N'Engineering', CAST(7236 AS Numeric(18, 0)), 3)

,(10, N'Pedro', N'Naldrett', N'Research and Development', CAST(5472 AS Numeric(18, 0)), 20)

,(11, N'Brina', N'Dillinger', N'Marketing', CAST(6512 AS Numeric(18, 0)), 2)

,(12, N'Verile', N'Sonley', N'Research and Development', CAST(4574 AS Numeric(18, 0)), 20)

,(13, N'Noble', N'Geerling', N'Research and Development', CAST(8391 AS Numeric(18, 0)), 20)

,(14, N'Garey', N'MacAdam', N'Accounting', CAST(3830 AS Numeric(18, 0)), 7)

,(15, N'Theo', N'Sorrell', N'Engineering', CAST(6442 AS Numeric(18, 0)), 3)

,(16, N'Erminie', N'Gelling', N'Research and Development', CAST(8591 AS Numeric(18, 0)), 20)

,(17, N'Loralie', N'Koop', N'Accounting', CAST(5248 AS Numeric(18, 0)), 7)

,(18, N'Cal', N'Andrey', N'Management Board', CAST(11259 AS Numeric(18, 0)), NULL)

,(19, N'Quincey', N'Gamell', N'Management Board', CAST(11367 AS Numeric(18, 0)), NULL)

,(20, N'Janith', N'McGiffie', N'Research and Development', CAST(7429 AS Numeric(18, 0)), 19)

,(18, N'Cal', N'Andrey', N'Management Board', CAST(11259 AS Numeric(18, 0)), NULL)

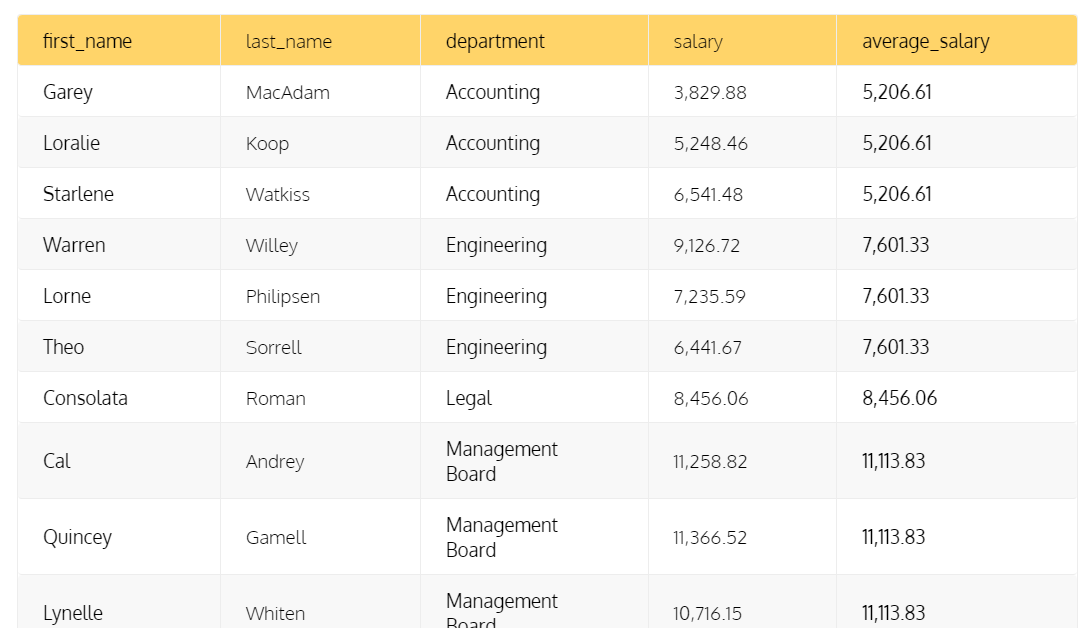
,(7, N'Starlene', N'Watkiss', N'Accounting', CAST(6541 AS Numeric(18, 0)), 4)

,(10, N'Pedro', N'Naldrett', N'Research and Development', CAST(5472 AS Numeric(18, 0)), 20)

,(8, N'Barde', N'Ribbens', N'Marketing', CAST(4853 AS Numeric(18, 0)), 2)

Q1. Using the table **employees**, show all the employees, their departments, salaries, and the average salary in their respective department. Order the result by department.

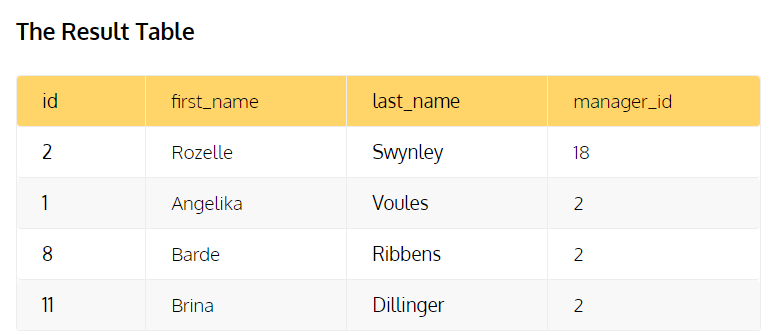
Sample Output:



Q2. Find the employee with the highest salary in each department. Show their first and last names, salaries, and departments.

Sample Output:

Q3. Find all employees working directly or indirectly under the employee whose ID is 18.



**Dataset - 2 (Sample Data only) (Q4 & Q5)**

DROP TABLE IF EXISTS dbo.Supporter

CREATE TABLE dbo.Supporter

(

Id int,

First\_Name varchar(50),

Last\_Name varchar(50)

)

INSERT INTO dbo.Supporter VALUES

(1,'Marlene','Wagner'),

(2,'Lonnie','Goodwin'),

(3,'Sophie','Peters'),

(4,'Edwin','Paul'),

(5,'Hugh','Thornton')

DROP TABLE IF EXISTS dbo.Projects

CREATE TABLE dbo.Projects

(

Id int,

Category varchar(50),

Author\_Id int,

Minimul\_Amount numeric

)

INSERT INTO dbo.Projects VALUES

(1,'Music',1,1677)

,(2,'Music',5,21573)

,(3,'Traveling',2,4952)

,(4,'Traveling',5,3135)

,(5,'Traveling',2,8555)

CREATE TABLE dbo.Donations

(

Id int,

Project\_Id int,

Supporter\_Id int,

Amount money,

Donated DATE

)

INSERT INTO dbo.Donations VALUES

(1,4,4,928.40,'2016-09-07')

,(2,8,18,384.38,'2016-12-16')

,(3,6,12,367.21,'2016-01-21')

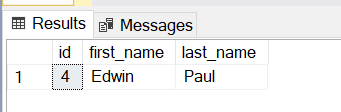
,(4,2,19,108.62,'2016-12-29')

,(5,10,20,842.58,'2016-11-30')

Q4. Obtain the project ID, minimal amount, and total donations for projects that have received donations over the minimum amount.

Q5. Select supporters who donated more than $200 total or who donated at least twice.

Sample Output



**Dataset for Q6 to Q10**

|  |  |
| --- | --- |
| **Customer** |  |
| **Id** | **PK** |
| **First\_Name** |  |
| **Last\_Name** |  |
| **Join\_Date** |  |
| **Country** |  |

|  |  |
| --- | --- |
| **Movie** |  |
| **Id** | **PK** |
| **Title** |  |
| **Release\_Year** |  |
| **Release\_Year** |  |
| **Editor\_Ranking** |  |

|  |  |
| --- | --- |
| **Review** |  |
| **Id** | **PK** |
| **Rating** |  |
| **Customer\_Id** | FK - Ref (Customer) |
| **Movie\_Id** | FK - Ref (Movie) |

|  |  |
| --- | --- |
| **Single\_Rental** |  |
| **Id** | **PK** |
| **Rental\_Date** |  |
| **Rental\_Period** |  |
| **Platform** |  |
| **Customer\_Id** | FK - Ref (Customer) |
| **Movie\_Id** | FK - Ref (Movie) |

|  |  |
| --- | --- |
| **Subscription** |  |
| **Id** |  |
| **Length** | In Days |
| **Start\_Date** |  |
| **Platform** |  |
| **Payment\_Date** |  |
| **Payment\_Amount** |  |
| **Customer\_Id** | FK - Ref (Customer) |

|  |  |
| --- | --- |
| **Giftcard** |  |
| **Id** |  |
| **Amount\_Worth** |  |
| **Customer\_Id** | FK - Ref (Customer) |
| **Payment\_Amount** |  |
| **Payment\_Date** |  |

Q6. For each single\_rental, show the rental\_date, the title of the movie rented, its genre, the payment amount, and the rank of the rental in terms of the price paid (the most expensive rental should have rank = 1). The ranking should be created separately for each movie genre. Allow the same rank for multiple rows and allow gaps in numbering.

Q7. Show the first and last name of the customer who bought the second most-recent gift card, along with the date when the payment took place. Assume that a unique rank is assigned for each gift card purchase.

Q8. Find all the customers who has given review to the movie. Show customer name (firstName, last\_Name), movie name and the corresponding rating.

Q9. Find no of movies taken on rental by each customer. Return movie name and customer name. Result should inlcude only those movies which was taken on rental for the current year

Q10. Find customer first\_name, last\_name for each platform who has been having active subscription for more than one platform.

Q11. Delete the duplicate records from table employees

.