/\*--TASK 16

--Task 16 **ROUGH WORK[DOUBLE CHECK FOR ERRORS]** - **work [ REMOVE $ SIGN AND THEN EXECUTE THE QUERIES.]**

for example the query

**SELECT \* FROM FactSales$;**

Should be typed as

**SELECT \* FROM FactSales; \*/**

RANK = IF FIRST RANK BY 2 PEOPLE, BOTH WILL BE GIVEN FIRST RANK AND THEN THEN THIRD RANK WILL B GIVN TO THE NEXT PERSON.

DENSE RANK = IF FIRST RANK BY 2 PEOPLE, BOTH WILL BE GIVEN FIRST RANK AND THEN THEN SECOND RANK WILL B GIVN TO THE NEXT PERSON.

\*/

--Task 16: Ranking

--The following examples show the four ranking functions used in the same query. Try the following queries and have a look at the results. What are the differences between them? Do they do what you expect?

--The DENSE\_RANK() is a window function that assigns a rank to each row within a partition of a result set.

--Unlike the RANK() function, the DENSE\_RANK() function returns consecutive rank values. Rows in each partition receive the same ranks if they have the same values.

SELECT EmployeeFirstName, EmployeeLastName

,ROW\_NUMBER() OVER (ORDER BY quantity) AS "Row Number"

,RANK() OVER (ORDER BY quantity) AS Rank

,DENSE\_RANK() OVER (ORDER BY quantity) AS "Dense Rank"

,PromotionType

,PromotionPercentage

FROM DimEmployee$ inner join FactSales$ on DimEmployee$.EmployeeID = FactSales$.EmployeeID inner join DimPromotion$ on FactSales$.PromotionID = DimPromotion$.PromotionID

where FactSales$.PromotionID IS NOT NULL;

SELECT EmployeeFirstName, EmployeeLastName

,ROW\_NUMBER() OVER (ORDER BY PromotionPercentage) AS "Row Number"

,RANK() OVER (ORDER BY PromotionPercentage) AS Rank

,DENSE\_RANK() OVER (ORDER BY PromotionPercentage) AS "Dense Rank"

,PromotionType

,PromotionPercentage

FROM DimEmployee$ inner join FactSales$ on DimEmployee$.EmployeeID = FactSales$.EmployeeID inner join DimPromotion$ on FactSales$.PromotionID = DimPromotion$.PromotionID

where FactSales$.PromotionID IS NOT NULL;