Coding Questions

1. Write a query that joins the STUDENT_AWARD and AWARD tables and displays the StudentID and AwardID (from the STUDENT_AWARD TABLE) and the CashPrize (from the AWARD TABLE). You will use this configuration for the rest of Window Function Queries.

```
--Write a query that joins the STUDENT_AWARD and AWARD tables and displays the StudentID and AwardID
-- (from the STUDENT_AWARD TABLE) and the CashPrize (from the AWARD TABLE).
-- You will use this configuration for the rest of Window Function queries.

SELECT sa.StudentID, sa.AwardID, a.CashPrize
FROM STUDENT_AWARD as sa JOIN AWARD as a
ON sa.AwardID = a.AwardID
```

100 %

Results | Messages

	StudentID	AwardID	CashPrize
1	1	1	80.00
2	1	2	70.00
3	2	1	80.00
4	5	1	80.00
5	7	1	80.00
6	10	4	22.00

2. Write an Aggregate Window Function that joins the STUDENT_AWARD and AWARD tables and creates a new column that includes the overall total cash prizes given. *The value for all rows in the new column should be the same (i.e., 412).

```
--Write an Aggregate Window Function that joins the STUDENT_AWARD and AWARD tables and
--creates a new column that includes the overall total cash prizes given.
-- *The value for all rows in the new column should be the same (i.e., 412).

SELECT sa.StudentID, sa.AwardID, a.CashPrize, SUM(a.CashPrize)
       OVER(Order by CashPrize) as [total]
FROM STUDENT_AWARD as sa JOIN AWARD as a
ON sa.AwardID = a.AwardID
```

100 %

Results | Messages

	StudentID	AwardID	CashPrize	total
1	10	4	22.00	22.00
2	1	2	70.00	92.00
3	1	1	80.00	412.00
4	2	1	80.00	412.00
5	5	1	80.00	412.00
6	7	1	80.00	412.00

3. Write an Aggregate Window Function that joins the STUDENT_AWARD and AWARD tables and creates a new column that includes the total cash prize earned for each student that has received an award. For this problem, use Partition, but not ORDER BY in the Window Function.

```
--Write an Aggregate Window Function that joins the STUDENT_AWARD and AWARD tables
-- and creates a new column that includes the total cash prize earned for each student
-- that has received an award. For this problem, use Partition,
-- but not ORDER BY in the Window Function.


SELECT sa.StudentID, sa.AwardID, a.CashPrize, SUM(a.CashPrize)
       OVER(Partition by CashPrize) as [total cash prize]
FROM STUDENT_AWARD as sa JOIN AWARD as a
ON sa.AwardID = a.AwardID
```

100 %

Results | Messages

	StudentID	AwardID	CashPrize	total cash prize
1	10	4	22.00	22.00
2	1	2	70.00	70.00
3	1	1	80.00	320.00
4	2	1	80.00	320.00
5	5	1	80.00	320.00
6	7	1	80.00	320.00

4. Repeat question 3, but this time include a running total for each student using the ORDER BY clause.

```
-- Repeat question 3, but this time include a running total for each student using
-- the ORDER BY clause.

SELECT sa.StudentID, sa.AwardID, a.CashPrize, SUM(a.CashPrize)
        OVER(Partition by
        StudentID
        Order by sa.AwardID) as [Running Total]
 FROM STUDENT_AWARD as sa JOIN AWARD as a
 ON sa.AwardID = a.AwardID
```

100 %

Results | Messages

	StudentID	AwardID	CashPrize	Running Total
1	1	1	80.00	80.00
2	1	2	70.00	150.00
3	2	1	80.00	80.00
4	5	1	80.00	80.00
5	7	1	80.00	80.00
6	10	4	22.00	22.00

5. **What is the difference in output between questions 3 and 4?**

Question three doesn't include a running total. It instead shows the total for each student.

6. Write Ranking Window Functions using RANK() based on the AWARD Table's CashPrize amounts.

```
--Write Ranking Window Functions using RANK() based on the AWARD Table's
--CashPrize amounts.


SELECT a.AwardID, a.CashPrize,
       Rank() OVER (Order By a.CashPrize DESC) AS [Olympics Rank]
FROM AWARD as a
```

100 %

Results | Messages

	AwardID	CashPrize	Olympics Rank
1	3	90.00	1
2	1	80.00	2
3	2	70.00	3
4	5	40.00	4
5	4	22.00	5

7. Write Ranking Window Functions using NTILE(4) based on the AWARD Table's CashPrize amounts.

```
-- Write Ranking Window Functions using NTILE(4) based on the AWARD Table's
-- CashPrize amounts.

SELECT a.AwardID, a.CashPrize,
       NTILE(4) OVER (Order By a.CashPrize DESC) AS [Quartile Rank]
FROM AWARD as a
```

100 %

Results | Messages

	AwardID	CashPrize	Quartile Rank
1	3	90.00	1
2	1	80.00	1
3	2	70.00	2
4	5	40.00	3
5	4	22.00	4