Lab 1 - Single table queries (Chapter 2)

Part 1

1. Using the MAX Function, write an SQL statement that shows the customer's last name closest to Z. Include Aliases when appropriate.

```
SELECT MAX(LastName) As LastName
FROM CUSTOMER;
```

100 %

Results | Messages

	LastName
1	Stull

2. Using the SALE_ITEM Table, write an SQL statement that shows the SALEID and COMMISSION for each sale.

```sql
SELECT
    SALEID,
    (0.1 * (UnitPrice * Quantity)) AS 'Commission'
FROM sale_item
ORDER BY Commission DESC;
```

100 %

Results | Messages

	SALEID	Commission
1	2	300.000
2	3	150.000
3	4	150.000
4	1	150.000
5	6	130.000
6	4	60.000
7	5	30.000

3) Modify the prior question to only show the highest SaleID and Commission using the TOP Function.

```sql
SELECT TOP 1
    SALEID,
    (0.1 * (UnitPrice * Quantity)) AS 'Commission'
FROM sale_item
ORDER BY Commission DESC;
```

100 %

Results | Messages

	SALEID	Commission
1	2	300.000

4) Write an SQL statement that shows Gender, and the total number of men and women that work for DPC Antiques. Order the output by the total from highest to lowest. Include Aliases when appropriate.

```sql
SELECT
    Gender,
    COUNT(Gender) AS No_of_People
FROM Employee
GROUP BY Gender
ORDER BY No_of_People DESC;
```

100 %

Results | Messages

	Gender	No_of_People
1	Male	7
2	Female	2

5. Write an SQL statement that shows the average salary of each employee. Don't include salaries below 30000 in the calculation. Include Aliases where appropriate.
*Describe why this question doesn't need to be answered using the AVG(Salary) aggregate function.

```
SELECT
    EmployeeID,
    Salary
FROM Employee
WHERE Salary >= 30000;
```

100 %

Results | Messages

	EmployeeID	Salary
1	1	60000.00
2	2	100000.00
3	4	150000.00
4	5	200000.00
5	6	80000.00
6	7	180000.00
7	8	150000.00
8	9	70000.00

Each Employee has only 1 record for salary. So the lone salary amount is the average salary. We would use AVG(Salary) if an employee had more than 1 salary amount given.

6) Write an SQL statement that shows the state and average salary of each state. Don't include salaries below 40000 in the calculation. Include Aliases when appropriate. In addition, group the averages by State. Don't include groups with less than 2 employees.

```sql
-- 6) Write an SQL statement that shows the state and average salary of each state.
-- Don't include salaries below 40000 in the calculation.
 -- Include an Aliases when appropriate.  In addition, group the averages by State.
 -- Don't include groups with less than 2 employees.

SELECT State, AVG(SALARY) as Average_Salary
FROM Employee
WHERE Salary >= 40000
GROUP BY State
HAVING COUNT(EMPLOYEEID) >= 2
```

100 %

Results | Messages

	State	Average_Salary
1	CA	110000.000000
2	UT	65000.000000

7) Write an SQL statement that shows the Employees' maximum salary offered in each state. Don't include states (groups) with a maximum salary less than $75,000.

```sql
SELECT
    State,
    MAX(Salary) AS Max_Salary
FROM Employee
GROUP BY State
HAVING MAX(Salary) >= 75000;
```

100 %

Results | Messages

	State	Max_Salary
1	CA	150000.00
2	HI	180000.00
3	MT	200000.00
4	TX	150000.00

8) Using a GROUP BY Clause, write an SQL statement that shows the TOP 3 hometowns of Employees based on total salaries being paid. Use the TOP and ORDER BY Clause to solve.

```
SELECT TOP 3
    City,
    SUM(Salary) AS Total_Salary
FROM Employee
GROUP BY City
ORDER BY Total_Salary DESC;
```

100 %

Results | Messages

	City	Total_Salary
1	San Diego	334000.00
2	Great Falls	200000.00
3	Laie	180000.00

9) Write an SQL statement that illustrates whether men or women employees have higher average salaries. Format the average salary column to 'currency' (i.e, $34.55).

```
SELECT
    Gender,
    FORMAT(AVG(Salary), 'C') AS Avg_Salary
FROM Employee
GROUP BY Gender;
```

100 %

Results | Messages

	Gender	Avg_Salary
1	Female	$190,000.00
2	Male	$87,714.29

10) Write an SQL statement that shows employee IDs that have completed more than one class with a grade of B or better. Include Aliases when appropriate. *This is a unique question where you might include a HAVING Clause to treat each employee as a group.

```
-- 10) Write an SQL statement that shows employee IDs that have completed more than one class with a grade of B or better.
-- Include an Aliases when appropriate.
-- *This is a unique question where you might include a HAVING Clause to treat each employee as a group.

SELECT EmployeeID, Count(Grade) as Count_Grade
FROM EMPLOYEE_TRAINING
WHERE Grade = 'A' OR Grade = 'B'
GROUP BY EmployeeId
HAVING Count(Grade) >= 2
```

Results | Messages

	EmployeeID	Count_Grade
1	1	2

11) Write an SQL statement that shows each Grade and the latest date where someone was assigned a grade using MAX(CompletionDate) code in the SELECT statement of solution.

```
SELECT
    Grade,
    MAX(CompletionDate) AS LatestDate
FROM Employee_Training
GROUP BY Grade;
```

100 %

Results | Messages

	Grade	LatestDate
1	A	2020-01-25 00:00:00.000
2	B	2019-05-23 00:00:00.000
3	C	2019-05-23 00:00:00.000
4	D	2020-01-25 00:00:00.000

PART 2

1) Curb uses the Employee data for corporate presentations. He would like his data output to show the full spelling for state information rather than abbreviations. The database administrator is unwilling to accommodate this request. Help Curb by creating a view called vSTATE that contains the full name of the employee along with the full spelling of the state using a CASE EXPRESSION. Just include the states that are included in the Employee Table.

```sql
-- 1)

CREATE VIEW vSTATE AS

SELECT
    CONCAT(RTRIM(FirstName), ' ', LastName) AS FullName,
CASE State
WHEN 'UT' THEN 'Utah'
WHEN'CA' THEN 'California'
WHEN 'TX' THEN 'Texas'
WHEN 'MT' THEN 'Montana'
WHEN 'HI' THEN 'Hawaii'
END AS FullStateName
FROM Employee;

-- Run View
SELECT * FROM vSTATE;

-- 2)
```

100 %

Results | Messages

	FullName	FullStateName
1	Bob Mills	Utah
2	John Johnson	California
3	David Olsen	California
4	Rich Matsko	Texas
5	Beverly Olsen	Montana
6	Curb Wendorf	California
7	Amy Bishop	Hawaii
8	Joe Hinz	California
9	Mike Dewer	Utah

2) For internal use, the antique store would like to create a view that provides additional information regarding product quality. In the current product database, if the quality is 1, the actual quality is Very Good. If the quality is 2, the actual quality is About Uncirculated. Finally, if the quality is 3, the actual quality is Mint State.

Write an SQL view called vQuality that creates a new virtual column that includes actual quality descriptions. Show the ProductID, ProductName, and New Column.

```sql
CREATE VIEW vQuality AS
SELECT
    Productid,
    ProductName,
CASE QualityID
WHEN 1 THEN 'Very Good'
WHEN 2 THEN 'About Uncirculated'
WHEN 3 THEN 'Mint State'
END AS ActualQuality
FROM Product;

-- Run View
SELECT * FROM vQuality;
```

100 %

Results | Messages

	Productid	ProductName	ActualQuality
1	1	American Silver Eagle	Very Good
2	2	Maple Leaf	About Uncirculated
3	3	Kookaburra	Mint State
4	4	Blue Chair	Mint State
5	5	Yellow Chair	Very Good
6	6	Green Chair	Mint State
7	7	Broken Bench	Very Good
8	8	Swing	Mint State

3) Write a searched CASE Expression using the PRODUCT table that provides a new column that displays 'High Quality' if qualityID is > 1 and 'Not Sure' for any other possibilities (ELSE statement). Show the ProductName and New column.

```sql
SELECT
    ProductName,
CASE
WHEN QualityID > 1 THEN 'High Quality'
ELSE 'Not Sure'
END AS 'ProductQuality'
FROM Product;
```

100 %

Results | Messages

	ProductName	ProductQuality
1	American Silver Eagle	Not Sure
2	Maple Leaf	High Quality
3	Kookaburra	High Quality
4	Blue Chair	High Quality
5	Yellow Chair	Not Sure
6	Green Chair	High Quality
7	Broken Bench	Not Sure
8	Swing	High Quality

4) The following is a list of 4 students who have received their final marks for the year. The graduation ceremony is near and the students have to be awarded Graduation Certificates with the respective Awards written on them. For this, we need to input the correct award in front of each student's name. The award criteria is as follows:

- Distinction: **90 - 100**
- Outstanding: **80- 89**
- Merit: **70-79**
- Below 70: **Fail**

StudentID	StudentName	Marks
1	Peter	75
2	Micheal	88
3	Fasih	99

4	Jack	65

Create Table and Insert data:

```sql
CREATE TABLE STUDENT_MARKS(
StudentID Int NOT Null,
StudentName Char(30) Not Null,
Marks INT Null,
);

INSERT INTO STUDENT_MARKS(StudentID, StudentName, Marks)
VALUES
    (1, 'Peter', 75),
    (2, 'Micheal', 88),
    (3, 'Fasih', 99),
    (4, 'Jack', 65);
```

Solution:

```sql
SELECT *,
CASE
    WHEN Marks BETWEEN 90 AND 99 THEN 'Distinction'
    WHEN Marks BETWEEN 80 AND 89 THEN 'Outstanding'
    WHEN Marks BETWEEN 70 AND 79 THEN 'Merit'
    ELSE 'Fail' END AS 'Award'
FROM STUDENT_MARKS;
```

100 %

Results | Messages

	StudentID	StudentName	Marks	Award
1	1	Peter	75	Merit
2	2	Micheal	88	Outstanding
3	3	Fasih	99	Distinction
4	4	Jack	65	Fail

This is a searched Case Expression.

5) **How many Years have passed since the Magna Carta was signed? Don't hardwire today's date. Use Aliases when appropriate.**

Years passed:

```
-- 5)

SELECT DATEDIFF(year, '12150615', GetDate()) AS 'YearsPassed';
```

100 %

Results | Messages

	YearsPassed
1	806

6) **How many days has it been since Huntsman Hall opened at Utah State and Wharton Business School.**

For Wharton, you can include the year and make up the month and day. The output should include two columns and include aliases.

```
SELECT
    DATEDIFF(day, '20160316', GetDate()) AS 'USU',
    DATEDIFF(day, '20020524', GetDate()) AS 'WHARTON';
```

100 %

Results | Messages

	USU	WHARTON
1	2010	7055