This worksheet is based on the **Advanced_Diploma_DB** database (same database used in class and worksheets). In this worksheet you are to use the same database that you have used in class

1. **Write a query that will display the employee id, name and surname, salary and commission columns found in the employee table.**
   - The column headings should be changed to 'ID', 'Employee Name', 'Salary' and 'Commission %' respectively.
   - Your query should list all the rows who do not have a null value in the commission column.
   - Your result should be sorted as shown below

   Sample Output:

	ID	Employee Name	Salary	Commission %
1	164	Mattea Marvins	7200.00	0.10
2	165	David Lee	6800.00	0.10
3	166	Sundar Ande	6400.00	0.10
4	167	Amit Banda	6200.00	0.10
5	173	Sundita Kumar	6100.00	0.10
6	179	Charles Johnson	6200.00	0.10
7	171	William Smith	7400.00	0.15
8	172	Elizabeth Bates	7300.00	0.15
9	178	Kimberely Grant	7000.00	0.15
10	163	Danielle Greene	9500.00	0.15

   Table to be used: employees
   Number of rows: 35

2. **Write a query that will display all the columns in the departments table.**
   - Your result should return only those rows which have a game location id equal to either 1800, 2400, 2500 or 2700.
   - The results should be sorted using the largest location id first and then by the smallest department Id (in case of identical location id).

   Sample output:

	department_id	department_name	manager_id	location_id
1	70	Public Relations	204	2700
2	80	Sales	145	2500
3	40	Human Resources	203	2400
4	20	Marketing	201	1800

   Table to be used: department
   Number of rows: 4

3. **Write a query that will list the identification code of countries which have been linked to a location. If a country is allocated to multiple locations its identification code should be listed only once. The column alias should be *Used Location Country Id's.***

Sample Output:

	Used Location Country Id's
1	AU
2	BR
3	CA
4	CH
5	CN
6	DE
7	IN
8	IT
9	JP
10	MX
11	NL
12	SG
13	UK
14	US

Table to be used: locations
Number of rows: 14

4. **Write a query that will return one single column named 'Employee Emails' that has the following format:**

<Employee Name> <Employee Surname> has the following email: <email>

- You are to use a single-row function to concatenate the strings together.
- The profile name should be modified such that all its characters are in capital letters.
- You are to return only those rows whose surname starts with the letter 'S' or 'J'.
- You are to sort the result using the email column.

The complete output:

	Employee Emails
1	James Marlow has email : JAMRLOW
2	John Chen has email : JCHEN
3	Julia Dellinger has email : JDELLING
4	Jennifer Dilly has email : JDILLY
5	Jean Fleaur has email : JFLEAUR
6	Janette King has email : JKING
7	James Landry has email : JLANDRY
8	Jack Livingston has email : JLIVINGS
9	Jason Mallin has email : JMALLIN
10	Jose Manuel Urman has email : JMURMAN

Table to be used: employees

Total number of rows: 29

5. **Write a query that will return a single column named 'Details' which has the following format:**
   <First letter of Name>.<Surname> was hired in <month hired> of <year hired>
   - Your answer should display the first letter of the Name followed by a period (.) and then the Surname which should be followed by the string 'was hired in' and the month and year in which the employee was hired.
   - The column should be named Employee Details
   - It is important to return only those employees whose Surname starts with the letter 'M' and ends with the letter 'S' (do not use LIKE)
   - The person who has been employed the longest should be listed first.

   The complete output:

	Employee Details
1	S.Mavris was hired in June of 1994
2	R.Matos was hired in March of 1998
3	K.Mourgos was hired in November of 1999
4	M.Marvins was hired in January of 2000

   Table to be used: employees
   Total number of rows: 4

6. **The below query returns 3 columns (Name, Surname and Initial Comparison). The Initial Comparison column should return 'Same Initial Letters' if both the Name and Surname start with the same letter or 'Different Initial Letters' otherwise. You are to list only employees whose surname starts with M and sort using the last name column.**

   Sample output:

	first_name	last_name	Initial Comparison
1	Jason	Mallin	Different Initial Letters
2	Steven	Markle	Different Initial Letters
3	James	Marlow	Different Initial Letters
4	Mattea	Marvins	Same Initial Letters
5	Randall	Matos	Different Initial Letters
6	Susan	Mavris	Different Initial Letters
7	Samuel	McCain	Different Initial Letters
8	Allan	McEwen	Different Initial Letters
9	Irene	Mikkilineni	Different Initial Letters
10	Kevin	Mourgos	Different Initial Letters

   Table to be used: employees
   Total number of rows: 10

7. **Write a query that will display one single column named 'Job Grades with less than 10,000 Maximum Salary' with the following format:**

<Job Grades>'s highest possible salary is: <highest salary>

- It is important that only the job grades with a highest salary of less than 10,000 are returned.
- The job grade with the smallest maximum salary should be placed first.
- It is also important that the concatenation is not performed using the CONCAT single-row function.

Sample output:

	Job Grades with less than 10,000 Maximum Salary
1	A's highest possible salary is: 2999
2	B's highest possible salary is: 5999
3	C's highest possible salary is: 9999

Table to be used: job_grades

Number of rows: 3

8. **Write a query that will return two columns.**
   - The first column should display the hire date.
   - The second column should display the name of the month in case that the person was hired in 'January', 'February', 'March', 'April' or 'May'. If the person was not hired in either of these months, then the second column should include the word 'Other'.
   - It is important that only the employees with an employee _id from 100 to 120 are returned.
   - The second column should be used to sort the results as shown below.

Sample output:

	Hire Date	Month Hired
1	1998-02-05 00:00:00.000	February
2	1999-02-07 00:00:00.000	February
3	1993-01-13 00:00:00.000	January
4	1990-01-03 00:00:00.000	January
5	1998-03-07 00:00:00.000	March
6	1995-05-18 00:00:00.000	May
7	1991-05-21 00:00:00.000	May
8	1997-06-25 00:00:00.000	Other
9	1994-08-17 00:00:00.000	Other
10	1994-08-16 00:00:00.000	Other

Table to be used: employees

Number of rows: 21