T-SQL EXERCISES

Exercise 1)

There is a **STUDENT_CLASSES** table that has the classes a student attended, with the GPA the student scored in each class. This table has the following columns:

```
STUDENT_ID      INTEGER,
CLASS_ID        INTEGER,
CLASS_GPA       NUMERIC (10, 3)
```

STUDENT_ID and CLASS_ID together uniquely identify a row in the STUDENT_CLASSES table.

There is an empty **STUDENT_OVERALL_GPA** table that is to store the overall GPA each student has scored across all their classes. This table has the following columns:

```
STUDENT_ID      INT,
OVERALL_GPA     NUMERIC (10, 3)
```

STUDENT_ID uniquely identifies a row in the STUDENT_CLASSES table.

Write the SQL statement(s) to create STUDENT_OVERALL_GPA records from the data in STUDENT_CLASSES.

Exercise 2)

2.1) **Write a SQL that generates all dates between a start and an end date.** The date rows cannot be read from a database table.

E.g.: If the start date is '2020-01-01' and the end date is '2020-01-07', the SQL should display:
2020-01-01
2020-01-02
2020-01-03
2020-01-04
2020-01-05
2020-01-06
2020-01-07

2.2) **Can you do this without using a loop?**

T-SQL EXERCISES

Exercise 3)

There is an **ITEM_INVENTORY** table that has the following columns:

```
Item_Inventory_ID    INTEGER
Item_ID              INTEGER
Inventory_DateTime   DATETIME
Inventory_Quantity   INTEGER
```

Item_Inventory_ID is the primary key for the table, which is based on a database sequence.
Item_ID identifies the items in the inventory.
Inventory_DateTime is the date and time when the inventory is taken.
Inventory_Quantity is the quantity of an item at a particular date and time.
Item_ID and Inventory_DateTime together form a unique index to the table.

Inventory is taken every hour and rows are added to ITEM_INVENTORY for all items. The Inventory_Quantity may or may not change each time the inventory is taken, depending on whether any sale occurred or not for that item over the past hour.

Some sample rows from ITEM_INVENTORY:

Item_Inventory_ID	Item_ID	Inventory_DateTime	Inventory_Quantity
1	100	2020-01-01 7AM	10
2	200	2020-01-01 7AM	20
3	100	2020-01-01 8AM	10
4	200	2020-01-01 8AM	15
5	100	2020-01-01 9AM	0
6	200	2020-01-01 9AM	5

3.1) **You need to determine the sales that occurred per hour for the last 24 hours for each item.** It needs to output rows that have Item_ID, hour and sales quantity for that item corresponding to the hour.
Using the above example, it would output:

Item_ID	Sale_Hour	Sale_Quantity
100	2020-01-01 7AM	0
100	2020-01-01 8AM	10
200	2020-01-01 7AM	5
200	2020-01-01 8AM	10

Can you do this for an item with a single SELECT statement (without explicitly looping through the sales for each hour)? If not, write a procedure that will determine the sales.

3.2) If the inventory was skipped for some reason during a particular hour, can we ensure that the missed hour still shows up in the report, with either zero sales or some indication of the missed reading? **Please explain if / how you could do this?**