Name: Pierre Augustamar

Class: INFX563

Question: What built-in functionality (including the data dictionary views) can be used in SQL Server to find out which indexes have been recently used?

Answer: To get info about the usage done against a particulate index. I will use the view sys.dm_db_index_usage_stats. When an index is used, a row is added to sys.dm_db_index_usage_stats and the counter for that row is incremented. There are several columns returned from this view, but the below columns provide helpful info about the index usage. The columns are:

- user_seeks - number of index seeks
- user_scans- number of index scans
- user_lookups - number of index lookups
- user_updates - number of insert, update or delete operations

```
GO
SELECT DISTINCT OBJECT_NAME(sis.OBJECT_ID) TableName,
       si.name AS IndexName, sc.Name AS ColumnName,
           sic.Index_ID, sis.user_seeks, sis.user_scans, sis.user_lookups, sis.user_updates
FROM sys.dm_db_index_usage_stats sis
                INNER JOIN sys.indexes si ON sis.OBJECT_ID = si.OBJECT_ID AND sis.Index_ID = si.Index_ID
                INNER JOIN sys.index_columns sic ON sis.OBJECT_ID = sic.OBJECT_ID AND sic.Index_ID =
si.Index_ID
                INNER JOIN sys.columns sc ON sis.OBJECT_ID = sc.OBJECT_ID AND sic.Column_ID = sc.Column_ID
GO
```

Here is a subset output of the above query using the AdventureWorks database.

TableName	IndexName	ColumnName	Index_ID	user_seeks	user_scans	user_lookups	user_updates	last_user_update
Contact	PK_Contact_ContactID	ContactID	1	0	3	0	2	10/30/2017 21:39
Contact	PK_Contact_ContactID	ContactID	1	45	0	0	0	NULL

From the result above, the user_updates counter indicates that a change was made on the index that is either an insert, delete or update operations on the related table under AdventureWorks database.

Question: How can you tell which ones have *NOT* been used?

Answer: To tell which indexes have not been used, you only need to check the counters. For instance, if an index is not listed when running the query specified above, then this tells us that this index has not been used. Thus, to list out the actual index that have not been used, then one can execute a query similar to the following:

```
SELECT  si.name
FROM sys.indexes si
WHERE si.name NOT IN (
SELECT DISTINCT si.name
FROM sys.dm_db_index_usage_stats sis
              INNER JOIN sys.indexes si ON sis.OBJECT_ID = si.OBJECT_ID AND sis.Index_ID
= si.Index_ID
              INNER JOIN sys.index_columns sic ON sis.OBJECT_ID = sic.OBJECT_ID AND
sic.Index_ID = si.Index_ID
              INNER JOIN sys.columns sc ON sis.OBJECT_ID = sc.OBJECT_ID AND sic.Column_ID
= sc.Column_ID
)
AND type_desc not in ('XML', 'HEAP')
AND name like 'PK%' OR name like 'AK%' OR name like 'IX%'
GO
```

Question: How can you find out which tables don't have any indexes?

```
SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME NOT IN (
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS)
```

Answer: The INFORMATION_SCHEMA.TABLE_CONSTRAINTS contains a row for each table that contains an index. Thus, to find tables that do not contain an index then a query that list out all tables not listed when querying the INFORMATION_SCHEMA_TABLE_CONSTRAINT view should list out tables without an index.