The following outlines the generic requirements for naming conventions, coding style and form design to be applied all student assignments and projects. Additional or alternative language-specific requirements will be provided in specific courses if appropriate.

# Notations:

|  |  |  |
| --- | --- | --- |
| **Database Container Objects** | | |
| **Level 1 Tags** | | |
| ***Hungarian*** *Notation with* ***CamelCase.*** *The first three letters identify the object type followed by a camelCase descriptive name.* | | |
| **Object** | **Prefix** | **Example(s)** |
| Table |  | Customers |
| View | vw | vwOverAchievers |
| Stored Procedure | sp | spCustomerList |
| Report | rpt | rptInsuranceValue |
| Triggers | trg | trgUpdateInventory |
| Field Names |  | customername customer\_name CustomerName |
|  |  |  |
| **Optional Level 2 Tags** | | |
| ***Hungarian*** *Notation with* ***CamelCase.*** *The first three letters identify the object type followed by a camelCase descriptive name.* | | |
| **Object** | **Prefix** | **Example** |
| Table | tbl | tblCustomer |
| Table (lookup) | tlkp | tlkpProvinces |
| Table (junction) | tjnc | tjncStudentCourses |
|  |  |  |
| **Constraint Naming** | | |
| ***Hungarian*** *Notation with* ***CamelCase.*** *The first three letters identify the object type followed by a camelCase descriptive name.* | | |
| **Object** | **Suffix** | **Example** |
| Primark Key | pk | customers\_pk |
| Foreign Key | fk | customer\_order\_fk |
| Required | req | customer\_name\_req |
| Unique | unq | customer\_email\_unq |
| Check | chk | customer\_yearOfBirth\_chk |
| Default | def | customer\_balance\_def |
| Index | idx | customer\_phone\_idx |

# SQL Files

When creating SQL files while using database management systems, such as Oracle, a standard single file format is to be created. Files shall include:

* All files should formatted in a way that they run without error (i.e. comment out any lines that are not functioning code.)
* A comment header that includes the author’s name, author’s id number, the date, purpose of the file, and a description of the file.
* Each question or separate entity shall include a comment description that includes the reference number and a brief sentence description.
* Descriptions and purposes should be in properly formatted English (i.e. spelling and grammar).

## Case Usage and Carriage Returns

* All **keyword commands** are to be in ALL-CAPS.   
  **Including**: SELECT, FROM, WHERE, HAVING, ORDER BY, GROUP BY, JOIN, CREATE, TABLE, VIEW, INSERT, UPDATE, VALUES, INTO, DELETE, DROP, ALTER, CONSTRAINT, AS, CHECK, INDEX, UNIQUE, DEFAULT, NULL, MODIFY, INDEX, FOREIGN KEY, PRIMARY KEY, COLUMN, NOT, AND, OR, REFERENCES, BETWEEN, DESCRIBE, ASC, DESC, ON, USING, IN, ANY, ALL
* All **single and multi-line functions** are to be in lower-case.   
  *example*: SELECT count(studentid) FROM tblStudents
* All **table and field names** are to be in lower case (pascal case is acceptable).
* **Carriage returns** should proceed
  + each main section of code (i.e. SELECT, FROM, JOIN, GROUP BY, WHERE, HAVING, ORDER BY
  + each subsidiary line of SQL should be indented (4 space
  + sub-queries should be further indented

# SQL Course Requirements

* Although the use of **regular expressions** in SQL can solve many issues and it is encouraged that they are explored, they are not part of the course and are **not to be used** in any submitted work.
* All SQL code written, is to be written in such a way that it would work **yesterday, today and tomorrow**. Data and data entry often change over time, SQL code should be written to still work with future data not yet entered.
* All SQL must be written considering **international date formats**. (i.e. The written SQL must work in both Canada and the United States or for various computer environmental settings)
* All SQL must be written **case agnostic**. Do not assume the data is entered correctly in the database and the SQL code must still function as intended regardless of the entered data’s case.  
  example: SELECT \* FROM students WHERE firstName = ‘Andrew’; assumes that the first letter is capitalized.

# Example:

----------------------------------------------------------------

-- Author: student’s name  
-- ID#: 100######  
-- Date: mmm dd, yyyy

-- Purpose: Submission for assignment # for course code

-- Description: This file includes several SQL scripts that

-- demonstrate the ability to retrieve data from a

-- relational database.

----------------------------------------------------------------

----------------------------------------------------------------  
-- Q1 – Display student names and ID numbers sorted by last name

SELECT lastname, firstname, studentnumber

FROM tblStudents

WHERE studentnumber BETWEEN 10000000 AND 99999999

ORDER BY lastname ASC;

/\* ----------------------------------------------------------------  
 Q2 – Display student first and last name as a single

field with an alias \*/

SELECT firstname || ‘ ‘ || lastname AS “Name”  
 FROM tblStudents

ORDER BY lastname ASC:

----------------------------------------------------------------  
-- END OF FILE

----------------------------------------------------------------