Skin Cancer Audit and
Research Data Inc
ABN 65506139325
QLD IA4670898

SPECIFICATION FOR THE UID - VERSION 2 **1ST AUGUST 2023**

Introduction

The main goal of deidentification is to protect a person's identity from being disclosed while still enabling the grouping of records related to that person.

The initial approach used a formula based on a person's name, which included using the first five letters of the name, date of birth, and sex. This resulted in a 14-character identifier for everyone based on the UID Version 1 Algorithm.

To import records into the SCARD Surgical Audit, this UID was the primary method of matching deidentified records to a person without the need of manual entry.

UID Collision

As the number of records in SCARD databases increased, UID "collisions" have arisen where multiple people can be assigned the same UID due to the limited characters used when calculating the identifier. Automated record importing involving duplicate UIDs is prevented and is deferred for manual entry.

SCARD UID Version 2 (UIDv2)

To address the issue of collisions, Version 2 of the SCARD UID formula was introduced. It utilizes 5 characters for both the first and last names and obfuscates the date of birth as hexadecimal, resulting in an 18-character identifier per person.

The method used by SCARD employs a concatemer of four demographic variables as an upper-case string:

First group (5 characters): The 1st, 2nd, 3rd, 5th and last letters of the individual's last name.

Second group (5 characters): The 1st, 2nd, 3rd, 5th and last letters of the individual's first name.

Third group (7 characters): The birth date as YYYYMMDD converted to hexadecimal.

Fourth group (1 character): The sex as an integer based on ISO/IEC 5218:
`0 – Not known, 1 - Male, 2 - Female, 9 – Not Applicable`

These groups are then combined into an 18-charater string. The result cannot be reverse engineered into a person's name as it lacks enough information; however, it still allows matching records during data transfers.

For example:

Name: DUSTY, Slim **DOB:** 1927-06-13 **Sex:** Male **UID:** UYSYDLMI2S1260BD51

Example Code and Functions

The GitHub repository at https://github.com/SCARDInc/UIDv2 contains examples and full functions for MySQL, MS SQL and PHP. This codebase will be expanded depending on the languages required.

Please note that MS SQL is not our native database language however working examples are provided.

Generating the UID as four Groups

First and Second Groups: First and Last Names

Both first and second groups use characters from the first name and last name. Any apostrophise, accents or other punctuation is ignored, and letters are converted to their Latin/English equivalent. These characters are used out of sequence to improve the randomness of the resulting string.

NOTE: If you are using a Unicode set with accented characters, these **MUST** be normalised to ASCII/Latin characters before generating output. Failure to follow this still may result in incorrect identifiers being generated across different platforms.

Character 1: The second letter of the name.

Character 2: The last letter of the name.

Character 3: The third letter of the name.

Character 4: The fifth letter of the name.

Character 5: The first letter of the name.

Short Names and substitution

For a shorter name, Character 2 is always the last letter of the name, even if is also the second or third letter of the name. As such in the case of a 5-letter name, both Character 2 and Character 4 will be the same.

If the specific character does not exist (such as a 2nd, 3rd or 5th letters), the missing letter(s) are substituted with the number 2. Following this rule, the name TA would appear as "AA22T".

Examples of name generation

BARTON: ANROB	**SMITH: MHIHS**	**WALLACE: AAELAW**	**LE BHERZ: EZBEL**
RUBY: UYB2B	**NG: GG22N**	**JON: ONN2J**	**ALLAN: LNLNA**

MySQL example for generating code for a single name.

```
SET @Name  = 'Ng';
SET @Name  = REGEX_REPLACE(@Name, '[^0-9a-zA-Z]', '');
SET @Char1 = IF(LENGTH(SUBSTRING(@Name, 2, 1)) > 0, SUBSTRING(@Name, 2, 1), '2');
SET @Char2 = SUBSTRING(@Name, -1, 1);
SET @Char3 = IF(LENGTH(SUBSTRING(@Name, 3, 1)) > 0, SUBSTRING(@Name, 3, 1), '2');
SET @Char4 = IF(LENGTH(SUBSTRING(@Name, 5, 1)) > 0, SUBSTRING(@Name, 5, 1), '2');
SET @Char5 = SUBSTRING @Name, 1, 1);

SELECT UCASE(CONCAT(@Char1,  @Char2,  @Char3,  @Char4,  @Char5)) AS `Name`;

Output Name: GG22N
```

MS SQL example for generating code for a single name.

NOTE 1: MS SQL is not our native SQL language and is provided as a working example.

NOTE 2: The MySQL example includes a Regex search to remove non-alphanumeric characters whereas MS SQL does not have a native feature. As such, you MUST strip non-alphanumeric characters before using the example script below to generate the codes.

```
DECLARE @Name VARCHAR (100) = 'Ng';
DECLARE @Char1 CHAR (1) = '2';
DECLARE @Char2 CHAR (1) = RIGHT(@Name, 1);
DECLARE @Char3 CHAR (1) = '2';
DECLARE @Char4 CHAR (1) = '2';
DECLARE @Char5 CHAR (1) = SUBSTRING(@Name, 1, 1);

IF LEN(@LastName) >= 2
    SET @Char1 = SUBSTRING(@LastName, 2, 1);

IF LEN(@LastName) >= 3
    SET @Char3 = SUBSTRING(@LastName, 3, 1);

IF LEN(@LastName) >= 5
    SET @Char4 = SUBSTRING(@LastName, 5, 1);

SELECT UPPER(CONCAT(@Char1, @Char2, @Char3, @Char4, @Char5)) AS 'Name';

Output Name: GG22N
```

Third Group: Date of birth

The third group takes the date of birth in the ISO8601 format (YYYY-MM-DD) but without separators (- or /). The resulting number is converted to an unsigned hexadecimal to add a layer of obfuscation to the date of birth. To find the hexadecimal value, the decimal number is divided by 16 until the quotient becomes zero.

Date of birth (ISO8601): 1982-01-25

Date of birth (without separators): 19820125

Converted to hexadecimal: 12E6E5D

Examples of date generation

1954-09-27: 12A2BBF **1985-05-07: 12EE50B** **1966-02-14: 12BFDB6** **2019-03-07: 1341463**

MySQL example for generating code for date of birth.

```
SET @Date = '1982-01-25';
SELECT UCASE(CONV(REGEX_REPLACE(@Date, '[^0-9]', ''), 10,16)) AS 'Date';

Outputs Date: 12E6E5D
```

MS SQL example for generating code for date of birth.

```
DECLARE @Date CHAR (10) = '1982-01-25';
SELECT FORMAT(CAST(REPLACE(@Date, '-', '') AS INT), 'X') AS 'Date';

Outputs Date: 12E6E5D
```

Forth Group: Sex

The fourth group is a single digit representing the sex of the person using codes from ISO/IEC 5218 (https://en.wikipedia.org/wiki/ISO/IEC_5218).

The four codes specified in ISO/IEC 5218 are:

- 0 = Not known
- 1 = Male
- 2 = Female
- 9 = Not applicable

Output of the UID

Combining all groups

Once all four groups have been generated, they are concatenated into a single uppercase string of 18 characters.

	Original	**Calculated**
Last Name:	DUSTY	UYSYD
First Name:	Slim	LMI2S
Date of Birth:	1927-06-13	1260BD5
Sex:	Male	1

Complete UID: UYSYDLMI2S1260BD51

Examples

HAWKE, Bob	**DOB:** 1929-05-16	**Sex:** Male	**UID:** AEWEHOBB2B12659941
SCHWARZENEGGER, Arnold	**DOB:** 1947-07-30	**Sex:** Male	**UID:** CRHASRDNLA129198A1
HAMILTON, Linda	**DOB:** 1956-09-27	**Sex:** Female	**UID:** ANMLHIANAL12A79DF2
ONO, Yoko	**DOB:** 1933-02-18	**Sex:** Female	**UID:** NOO2OOOK2Y126F4AA1

End of document