



Универзитет „Св. Кирил и Методиј“ во Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Тема:

Анонимизација на оперативна база на податоци

Предмет:

Складишта на податоци и аналитичка обработка

Изработила:

Душица Јанковиќ (161509)

Август, 2020

Задача

За произволна база, да се генерираат SELECT наредби кои ги маскираат колоните кои се означени како Лични или Заштитени (пр. матичен број, име и презиме, состојба на трансакциска сметка, итн.). Некои од нив ќе треба да се маскираат со други вредности, некои да се изостават, итн.

База на податоци

За целите на овој проект тестирав со локално ресторирана верзија на AdventureWorks2012 базата на податоци.

Влез

На влез се вчитуваат информации за базата на податоци. Информациите се поделени на два дела.

1. Првиот документ служи на опишување на секоја табела. Тој е tab delimited и колоните се table name, column name, datatype, nullable, anonymization type, default value. На следната слика се прикажани првите 20 реда (input_properties.txt).

TABLE_NAME	COLUMN_NAME	DATATYPE	NULLABLE	ANONYMIZATION	DEFAULT VALUE
HumanResources.Department	DepartmentID	smallint	NOT NULL	RandomFromSet	
HumanResources.Department	Name	nvarchar(50)	NOT NULL	Omit	
HumanResources.Department	GroupName	nvarchar(50)	NOT NULL	RandomPseudonymFromSet	
HumanResources.Department	ModifiedDate	datetime	NOT NULL	Omit	
HumanResources.Employee	BusinessEntityID	int	NOT NULL	RandomFromSet	
HumanResources.Employee	NationalIDNumber	nvarchar(15)	NOT NULL	RandomPseudonym	
HumanResources.Employee	LoginID	nvarchar(256)	NOT NULL	RandomPseudonym	
HumanResources.Employee	OrganizationNode	hierarchyid	NULL	Random	
HumanResources.Employee	OrganizationLevel	smallint	NULL	Random	
HumanResources.Employee	JobTitle	nvarchar(50)	NOT NULL	RandomPseudonym	
HumanResources.Employee	BirthDate	date	NOT NULL	Random	
HumanResources.Employee	MaritalStatus	nchar(1)	NOT NULL	RandomPseudonymFromSet	
HumanResources.Employee	Gender	nchar(1)	NOT NULL	RandomFromSet	
HumanResources.Employee	HireDate	date	NOT NULL	Random	
HumanResources.Employee	SalariedFlag	bit	NOT NULL	Random	
HumanResources.Employee	VacationHours	smallint	NOT NULL	None	
HumanResources.Employee	SickLeaveHours	smallint	NOT NULL	Random	
HumanResources.Employee	CurrentFlag	bit	NOT NULL	Default value	0
HumanResources.Employee	rowguid	uniqueidentifie	NOT NULL	Default value	
HumanResources.Employee	ModifiedDate	datetime	NOT NULL	Default value	

2. Вториот документ има информации за надворешни клучеви. Тој вклучува четири колони referencing table, referencing column, referenced table, referenced column. Првите две колони како пар дефинираат од каде е надворешниот клуч, вторите две дефинираат до што покажува надворешниот клуч. Ќе се ограничине на прости надворешни клучеви составени само од една колона. На следната слика се прикажани неколку редови од овој документ (input_references.txt).

REFERENCING_TABLE	REFERENCING_COLUMN	REFERENCED_TABLE	REFERENCED_COLUMN
Person.Address	StateProvinceID	Person.StateProvince	StateProvinceID
Production.BillOfMaterials	ProductAssemblyID	Production.Product	ProductID
Production.BillOfMaterials	ComponentID	Production.Product	ProductID
Production.BillOfMaterials	UnitMeasureCode	Production.UnitMeasure	UnitMeasureCode
Person.BusinessEntityAddress	AddressID	Person.Address	AddressID
Person.BusinessEntityAddress	AddressTypeID	Person.AddressType	AddressTypeID
Person.BusinessEntityAddress	BusinessEntityID	Person.BusinessEntity	BusinessEntityID
Person.BusinessEntityContact	PersonID	Person.Person	BusinessEntityID
Person.BusinessEntityContact	ContactTypeID	Person.ContactType	ContactTypeID
Person.BusinessEntityContact	BusinessEntityID	Person.BusinessEntity	BusinessEntityID
Sales.CountryRegionCurrency	CountryRegionCode	Person.CountryRegion	CountryRegionCode
Sales.CountryRegionCurrency	CurrencyCode	Sales.Currency	CurrencyCode
Sales.CurrencyRate	FromCurrencyCode	Sales.Currency	CurrencyCode
Sales.CurrencyRate	ToCurrencyCode	Sales.Currency	CurrencyCode
Sales.Customer	PersonID	Person.Person	BusinessEntityID
Sales.Customer	StoreID	Sales.Store	BusinessEntityID
Sales.Customer	TerritoryID	Sales.SalesTerritory	TerritoryID
Production.Document	Owner	HumanResources.Employee	BusinessEntityID

Анонимизација

Возможни се неколку типови на анонимизација:

1. **None** - колоната не се менува, останува каква што си е во оперативната база.
2. **Omit** - колоната тотално се отстранува од SELECT наредбата
3. **Default value** - вредноста што била доделена во default constraint или проследена во конфигурацијата на анонимизацијата
4. **Set Null** - се поставува null вредност за таа колона. Доколку колоната не е nullable се враќа грешка.
5. **Random** - случајна вредност, со тоа што два пати ако се повика за иста вредност се добива различна анонимизирана вредност(пр. 1 -> 123, вториот пат 1 -> 234)
6. **RandomPseudonym** - случајна вредност, со тоа што два пати ако се повика за иста вредност (пр. 1 -> 123, вториот пат исто така 1 -> 123)
7. **RandomPseudonymPerTable** - исто како предходното, но мапирањата се 'памтат' само на ниво на една табела. Ова значи ако вредноста 1 треба постојано да се мапира во 123 во табела A, не значи дека вредност 1 во друга табела треба да го почитува истото мапирање.
8. **RandomFromSet** - да се одбере случајна вредност од множество на вредности. Доколку колоната на која се применува оваа анонимизација е надворешен клуч кон друга табела, тогаш множеството вредности во кои ќе може да се мапира колоната се превзема од референцираната табела. Доколку колоната не е надворешен клуч, тогаш множеството дозволени вредности се составува од сите вредности кои се јавуваат за таа колона во тековната табела.
9. **RandomPseudonymFromSet** - исто како и предходната анонимизација, но се води сметка за мапирањата така што една вредност треба секогаш да се мапира во иста вредност од дозволеното множество.

Имплемент¹ација

Во овој проект користев python за вчитување и парсирање на влезните фајлови, а потоа и за генерирање на SQL наредби зависно од конфигурацијата во влезните фајлови. Главната цел на овој проект е за база на податоци со табели А, Б .. да се генерираат анонимизирани views именувани според шемата vw_A_Anonymized, vw_B_Anonymized... соодветно.

Влез

Податоците се читаат од два фајлови со структура објаснета погоре. Локацијата на овие документи се праќаат како command line аргументи или пак се читаат од input датотеката. За вчитување и парсирање на податоците е задолжен фајлот [anonymize_from_input.py](#), кој подоцна се повикува на random_generators.py за да ја изврши анонимизацијата.

Анонимизација

Целата логика за генерирањето на DDL и DML наредби потребни за анонимизација и помошни табели и views е во [random_generators.py](#). Следно ќе ги разгледаме функциите и како тие работат заедно за да ги остварат сите 9 анонимизациски типа.

За **omit** и **none** не се потребни посебни функции бидејќи анонимизацијата тука е едноставна, или тотално ја испушта колоната, или пак ја остава каква што е оригинално. Доколку сите колони од една табела се анонимизираат со опцијата **omit** тогаш за таа табела воопшто не се генерира анонимизиран view.

Анонимизацијата **set null** е исто така едноставна, се што треба е да се прикаже NULL наместо вредноста која всушност е во табелата. Тука се прави дополнителна проверка за тоа дали дадена табела е nullable. Ова се прави со пребарување и споредба со вредноста за nullable во влезните фајлови.

Set default анонимизацијата е замена на вредност во колоната со препратената вредност во конфигурацискиот влезен фајл. Доколку не е препратена никаква вредност, тогаш вредноста се влече од column_default за колоната во таа табела. Оваа информација се добива од *information_schema.columns* табелата. Тука е важно да се фрли грешка ако влезниот фајл нема наведено default value, а ни column_default (вредноста за таа колона е null - но не како вредност туку за да симболизира отсуство на информација за default вредност). Со if else блок се прават овие проверки.

- [get_defaultValueRetrieval_statement](#) е функција со која се влече column_default од information_schema.columns

¹ <https://github.com/djankovik/Anonymization-Project-SnPiAO>

За генерирање на **random** вредност за било кој податочен тип е имплементирана функцијата *getRND_generating_statement_for_datatype(datatype)* - е функција која за даден препратен податочен тип како string враќа наредба за генерирање на рандом вредност од тој податочен тип. За секој податочен тип (или група типови) се повикува соодветна функција за генерирање SQL израз. Во прилог се дадени функциите за секој податочен тип, а под нив е дадена една пример наредба која би се генерирала:

- *generateXInt_statement*: tinyint, smallint, int, bigint
`CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(10000)) AS SMALLINT)`
- *generateRandomMoney_statement*: money, smallmoney
`CAST((RAND(CHECKSUM(NEWID()))*(100000)) AS MONEY)`
- *generateRandomNumeric_or_Decimal_statement*: numeric(p,s) and decimal(p,s) for any precision and scale arguments
`CAST((RAND(CHECKSUM(NEWID()))*(1000000)) AS DECIMAL(6,4))`
- *generateRealorFloat_statement*: real and float
`CAST((RAND(CHECKSUM(NEWID()))*(1000000000)) AS FLOAT)`
- *generateUniqueIdentifier*: uniqueidentifier
`CAST(NEWID() AS UNIQUEIDENTIFIER)`
- *generateHierarchyId*: hierarchyid
`CONVERT(VARBINARY(5), CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(100000)) AS VARBINARY))`
- *generateRandomDateOrTime_statement*: date, time, datetime, timestamp
`CAST(DATEADD(SECOND, ROUND(((DATEDIFF(SECOND, '2000-01-01 00:00:00', '2020-01-01 00:00:00')-1) * RAND(CHECKSUM(NEWID()))), 0), '2000-01-01 00:00:00') AS DATETIME)`
- *generateRandomXChar_statement*: char, nchar, varchar, nvarchar for any length.
`CAST(dbo.GenerateRandomString() AS NVARCHAR(100))`
- *generateRandomXBinary_statement*: binary and varbinary
`CAST(CRYPT_GEN_RANDOM(8000) AS VARBINARY(50))`
- *generateRandomBit_statement*: bit
`CAST(ROUND(RAND(CHECKSUM(NEWID()))),0) AS BIT)`

Функцијата *GenerateRandomString* е функција која генерира random string со максимална должина кој подоцна се cast-ира во помал зависно од потребите.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION GenerateRandomString
RETURNS nvarchar(4000)
AS
BEGIN
DECLARE @length int,
        @charpool nvarchar(100),
        @poollength int,
        @loopcount int,
        @randomstring nvarchar(4000);
SET @length = 4000
SET @CharPool =
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789, . ! ? ; - _ '
SET @PoolLength = Len(@CharPool)
```

За анонимизацијата **generate random from set** е имплементирана функцијата *getRandomFromSet(tablename, columnname, references_dict)* која зависно од тоа дали колоната е

надворешен клуч или не ги селектира сите distinct вредности од референцираната табела или тековната табела соодветно.

Анонимизациите **random_pseudonym**, **random_pseudonym per table** и **random_pseudonym from set** ги користат функциите споменати погоре, но дополнително се креираат и табели за да се чуваат мапирањата оригинална вредност → анонимизирана вредност. За чување на мапирањата прво се креираат табелите каде ќе бидат запишани мапирањата како fromvalue и tovalue колони од соодветен податочен тип, а потоа се полнат овие табели со процедури. Попрецизно за секој анонимизациски тип:

- ***RandomPseudonym***: За секој податочен тип се генерираат мапирања. Станува збор за мапирање на глобално ниво односно: за вредност A од податочен тип x доколку е дефинирано мапирање $A \rightarrow B$ тогаш било која колона од било која табела од податочен тип x ја мапира вредноста A (доколку ја содржи) во B.
 - Креирање на табелите: *getPseudoRandomMappings_creation_statement*
 - Полнење: *getPseudoRandomMappings_fillup_procedure*
- ***RandomPseudonymPerTable***: Слично како горното, но наместо на глобално (на цела база на податоци), мапирањата се прават на табеларно ниво. Ова значи дека вредноста A ќе се мапира во B за таа табела, но во други табели за истата вредност A може да е дефинирано друго мапирање.
 - Креирање на табелите: *getPseudoRandomPerTableMappings_creation_statement*
 - Полнење: *getPseudoRandomPerTableMappings_fillup_procedure*
- ***RandomPseudonymFromSet***: Се влечат вредности од референцирана табела (ако колоната која се анонимизира е надворешен клуч) или од тековната табела.
 - Креирање на табелите: *getPseudoRandomFromSet_creation_statement*
 - Полнење: *getPseudoRandomFromSet_insertinto_statement*

Излез

Генерираните наредби за create и drop се видливи во датотеката generated_statements.

Пример

Оригинални vs Анонимизирани колони

Sales.SpecialOffer

	SpecialOfferID	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty	rowguid	ModifiedDate
1	1	No Discount	0.00	No Discount	No Discount	2011-05-01 00:00:00.000	2014-11-30 00:00:00.000	0	NULL	0290C4F5-191F-4337-AB6B-0A2DDE03CBF9	2011-04-01 00:00:00.000
2	2	Volume Discount 11 to 14	0.02	Volume Discount	Reseller	2011-05-31 00:00:00.000	2014-05-30 00:00:00.000	11	14	D7542EE7-15D8-4541-985C-5CC27AEF26D6	2011-05-01 00:00:00.000
3	3	Volume Discount 15 to 24	0.05	Volume Discount	Reseller	2011-05-31 00:00:00.000	2014-05-30 00:00:00.000	15	24	48DBCC01-8CF7-40A9-B643-40EC5B717491	2011-05-01 00:00:00.000
4	4	Volume Discount 25 to 40	0.10	Volume Discount	Reseller	2011-05-31 00:00:00.000	2014-05-30 00:00:00.000	25	40	504B5E85-8F3F-4EBC-9E1D-C1BC5DEA9A88	2011-05-01 00:00:00.000
5	5	Volume Discount 41 to 60	0.15	Volume Discount	Reseller	2011-05-31 00:00:00.000	2014-05-30 00:00:00.000	41	60	677E1D9D-944F-4E81-90E8-47EB0A82D48C	2011-05-01 00:00:00.000
6	6	Volume Discount over 60	0.20	Volume Discount	Reseller	2011-05-31 00:00:00.000	2014-05-30 00:00:00.000	61	NULL	8157F569-4E8D-46B6-9347-5D0F726A9439	2011-05-01 00:00:00.000
7	7	Mountain-100 Clearanc...	0.35	Discontinued P...	Reseller	2012-04-13 00:00:00.000	2012-05-29 00:00:00.000	0	NULL	7DF15BF5-6C05-47E7-80A4-22BD1CE59A72	2012-03-14 00:00:00.000
8	8	Sport Helmet Discount...	0.10	Seasonal Disc...	Reseller	2012-05-30 00:00:00.000	2012-06-29 00:00:00.000	0	NULL	20C5D2CC-A3F8-48F8-AC9A-8F15943E52AE	2012-04-30 00:00:00.000
9	9	Road-650 Overstock	0.30	Excess Inventory	Reseller	2012-05-30 00:00:00.000	2012-07-30 00:00:00.000	0	NULL	0CF8472B-F9E6-4945-9E09-549D7DDE2198	2012-04-30 00:00:00.000
10	10	Mountain Tire Sale	0.50	Excess Inventory	Customer	2013-05-14 00:00:00.000	2013-07-29 00:00:00.000	0	NULL	220444AD-2EF3-4E4C-87E9-3AA6EE39A877	2013-04-14 00:00:00.000

SpecialOfferID	Random
Description	Omit
DiscountPct	RandomPseudonymPerTable
Type	Omit
Category	RandomFromSet
StartDate	RandomPseudonym
EndDate	Omit
MinQty	RandomFromSet
MaxQty	Set Null
rowguid	Omit
ModifiedDate	Omit

	SpecialOfferID	DiscountPct	Category	StartDate	MinQty	MaxQty
1	809149523	64376.0495	No Discount	2019-03-27 21:42:07.000	15	NULL
2	363907423	85900.3678	Customer	2015-09-02 00:12:11.000	0	NULL
3	769399680	45296.8425	Customer	2015-09-02 00:12:11.000	11	NULL
4	240021115	84144.8961	Customer	2015-09-02 00:12:11.000	25	NULL
5	125393018	84144.8961	Customer	2007-11-15 14:52:26.000	15	NULL
6	620946828	93284.4569	Customer	2015-09-02 00:12:11.000	61	NULL
7	989098950	93284.4569	Customer	2016-02-22 12:22:12.000	15	NULL
8	401927976	93284.4569	Customer	2016-02-22 12:22:12.000	15	NULL
9	907086330	29308.9944	Customer	2015-09-02 00:12:11.000	41	NULL
10	264104019	29308.9944	Customer	2016-02-22 12:22:12.000	15	NULL

HumanResources.Shift

ShiftID	Name	StartTime	EndTime	ModifiedDate
1	Day	07:00:00.0000000	15:00:00.0000000	2008-04-30 00:00:00.000
2	Evening	15:00:00.0000000	23:00:00.0000000	2008-04-30 00:00:00.000
3	Night	23:00:00.0000000	07:00:00.0000000	2008-04-30 00:00:00.000

ShiftID	RandomPseudonymFromSet
Name	RandomFromSet
StartTime	RandomPseudonymPerTable
EndTime	Default value
ModifiedDate	Omit

ShiftID	Name	StartTime	EndTime
3	Day	14:27:15.0000000	2020-07-28 11:10:08.423
1	Night	07:23:17.0000000	2020-07-28 11:10:08.423
2	Evening	03:42:01.0000000	2020-07-28 11:10:08.423

HumanResources.Employee

BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate	MaritalStatus	Gender	HireDate	SalariedFlag	VacationHours	SickLeaveHours	CurrentFlag	rowguid	ModifiedDate
1	295047204	adventure-works\sa...	NULL	NULL	Chief Executive Officer	1969-01-29	S	M	2009-01-14	1	99	69	1	F01251E5-96A3-448D-9E1E-0F...	2014-06-30 00:00:00.000
2	245797967	adventure-works\sa...	0x58	1	Vice President of En...	1971-05-01	S	F	2008-01-31	1	20	1	1	49E8F437-670D-4409-90CB-F9...	2014-06-30 00:00:00.000
3	559647174	adventure-works\sa...	0x5ACD	2	Engineering Manager	1974-11-12	M	M	2007-11-11	1	2	21	1	98889B2C-6788-4217-5887-F9...	2014-06-30 00:00:00.000
4	112457891	adventure-works\sa...	0x5ADA	3	Senior Tool Designer	1974-12-23	S	M	2007-12-05	0	48	80	1	59747955-4785-443F-8E34-F8...	2014-06-30 00:00:00.000
5	695256908	adventure-works\sa...	0x5ADA	3	Design Engineer	1952-09-27	M	F	2008-01-06	1	5	22	1	EC8AAE09-F908-4415-B440-6C...	2014-06-30 00:00:00.000
6	998320692	adventure-works\sa...	0x5ADE	3	Design Engineer	1959-03-11	M	M	2008-01-24	1	6	23	1	E36056F1-9C05-478D-9945-14...	2014-06-30 00:00:00.000
7	134969118	adventure-works\sa...	0x5AE1	3	Research and Devel...	1987-02-24	M	M	2009-02-08	1	61	50	1	4F40DECA-EF01-41FD-8029-3A...	2014-06-30 00:00:00.000
8	811984146	adventure-works\sa...	0x5AE158	4	Research and Devel...	1986-06-05	S	F	2008-12-29	1	62	51	1	31112635-6638-4010-8442-A6...	2014-06-30 00:00:00.000
9	858797903	adventure-works\sa...	0x5AE168	4	Research and Devel...	1979-01-21	M	F	2009-01-16	1	63	51	1	5266C0C6-7570-47EF-9570-48...	2014-06-30 00:00:00.000
10	879342154	adventure-works\sa...	0x5AE178	4	Research and Devel...	1984-11-30	M	M	2009-05-03	1	16	64	1	EA43880D-5571-40CB-AB1A-3B...	2014-06-30 00:00:00.000

HumanResources.Employee	Random
HumanResources.Employee	Omit
HumanResources.Employee	Omit
HumanResources.Employee	Random
HumanResources.Employee	Set null
HumanResources.Employee	Omit
HumanResources.Employee	Random
HumanResources.Employee	RandomPseudonymFromSet
HumanResources.Employee	RandomFromSet
HumanResources.Employee	Omit
HumanResources.Employee	RandomPseudonym
HumanResources.Employee	Random
HumanResources.Employee	RandomPseudonym
HumanResources.Employee	Random
HumanResources.Employee	Default value
HumanResources.Employee	Default value

BusinessEntityID	OrganizationNode	OrganizationLevel	BirthDate	MaritalStatus	Gender	SalariedFlag	VacationHours	SickLeaveHours	CurrentFlag	rowguid	ModifiedDate
466582168	0x40E7414000	NULL	2011-05-22	S	F	1	7377	8384	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
471547103	0x40F40E9000	NULL	2008-03-17	S	M	1	5774	7353	1	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
974382515	0x40E9C50000	NULL	2012-04-10	M	F	1	5747	1737	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
265579775	0x40B0A70000	NULL	2017-09-14	S	F	0	6164	3639	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
327019719	0x40D1214000	NULL	2014-03-15	M	M	1	2140	8810	1	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
368302879	0x40E678C000	NULL	2009-01-08	M	F	1	5985	1549	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
125416476	0x40A1EAD000	NULL	2012-10-01	M	F	1	1578	7172	1	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
683176052	0x40E7206000	NULL	2011-04-24	S	M	1	9561	5995	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
316543238	0x40D2B60000	NULL	2012-07-07	M	M	1	1901	5995	0	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000
914444524	0x40AC0C0000	NULL	2010-11-15	M	F	1	2727	9397	1	D7D15C85-AE57-48C4-8BC2-07441C804F33	2020-07-28 11:18:00.000

Генериран код

Креирање на VIEWS

```

USE ADVENTUREWORKS_1907
GO

CREATE VIEW vw_TablesColumnsProperties
AS
SELECT CONCAT(table_schema, '.', table_name) as schema_table_name, table_schema as
schema_name, table_name, column_name, data_type, is_nullable, column_default
FROM information_schema.columns
GO

CREATE VIEW vw_getRANDValue
AS
SELECT RAND() AS Value
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION GenerateRandomString()
RETURNS nvarchar(4000)
AS
BEGIN
DECLARE @length int, @charpool nvarchar(100), @poollength int, @loopcount int, @randomstring
nvarchar(4000);
SET @length = 4000
SET @CharPool = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789, .!?:; -_ '
SET @PoolLength = Len(@CharPool)
SET @LoopCount = 0
SET @RandomString = ''
WHILE (@LoopCount < @Length) BEGIN
    SELECT @RandomString = @RandomString +
    SUBSTRING(@Charpool, CONVERT(int, (SELECT Value FROM vw_getRANDValue) * @PoolLength), 1)
    SELECT @LoopCount = @LoopCount + 1
END
RETURN @RandomString
END
GO

go

create procedure sp_fillup_RandomPseudonymTableMap @mappingstable nvarchar(100), @tablename
nvarchar(100), @columnname nvarchar(100), @randomgenstatement nvarchar(4000)
as

```



```

begin set nocount on

    exec('insert into '+@mappingstable+'(tablename,columnname,fromvalue,tovalue)
        select distinct '''+@tablename+''', '''+@columnname+''', t.'+@columnname+',
'+@randomgenstatement+' as tovalue from (select distinct '+@columnname+' from '+@tablename+') t
        left outer join (select distinct fromvalue, tovalue from '+@mappingstable+' where tablename =
'''+@tablename+'''' and columnname='''+@columnname+''') dt
        on t.'+@columnname+' = dt.fromvalue where dt.tovalue is null')

end

go

create procedure sp_fillup_RandomPseudonymMap @mappingstable nvarchar(100), @tablename
nvarchar(100), @columnname nvarchar(100), @randomgenstatement nvarchar(4000)
as
begin
    set nocount on
    exec('insert into '+@mappingstable+'(fromvalue,tovalue)
        select distinct t.'+@columnname+', '+@randomgenstatement+'
        from (select distinct '+@columnname+' from '+@tablename+') t
        left outer join (select distinct fromvalue, tovalue from '+@mappingstable+') dt
        on t.'+@columnname+' = dt.fromvalue where dt.tovalue is null')

end

GO

CREATE TABLE bit_RandomPseudonymMap( fromvalue bit, tovalue bit);
exec sp_fillup_RandomPseudonymMap 'bit_RandomPseudonymMap', 'HumanResources.Employee',
'SalariedFlag', 'CAST(ROUND(RAND(CHECKSUM(NEWID()))),0) AS BIT)'

GO

CREATE TABLE smallint_RandomPseudonymMap( fromvalue smallint, tovalue smallint);
exec sp_fillup_RandomPseudonymMap 'smallint_RandomPseudonymMap', 'HumanResources.Employee',
'SickLeaveHours', 'CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(10000)) as SMALLINT)'

GO

CREATE TABLE datetime_RandomPseudonymMap( fromvalue datetime, tovalue datetime);
exec sp_fillup_RandomPseudonymMap 'datetime_RandomPseudonymMap', 'Sales.SpecialOffer', 'StartDate',
'CAST(CAST(DATEADD(SECOND, ROUND(((DATEDIFF(SECOND, ''2000-01-01 00:00:00'', ''2020-01-01
00:00:00'')-1) * RAND(CHECKSUM(NEWID()))), 0), ''2000-01-01 00:00:00'') AS datetime) as datetime)'

GO

CREATE TABLE time_7_RandomPseudonymPerTableMap(tablename nvarchar(100),columnname nvarchar(100),
fromvalue time(7), tovalue time(7));
exec sp_fillup_RandomPseudonymTableMap 'time_7_RandomPseudonymPerTableMap', 'HumanResources.Shift',
'StartTime', 'CAST(CAST(DATEADD(SECOND, ROUND(((DATEDIFF(SECOND, ''2000-01-01 00:00:00'',
''2020-01-01 00:00:00'')-1) * RAND(CHECKSUM(NEWID()))), 0), ''2000-01-01 00:00:00'') AS datetime) as
time(7))'

GO

CREATE TABLE smallmoney_RandomPseudonymPerTableMap(tablename nvarchar(100),columnname nvarchar(100),
fromvalue smallmoney, tovalue smallmoney);
exec sp_fillup_RandomPseudonymTableMap 'smallmoney_RandomPseudonymPerTableMap',
'Sales.SpecialOffer', 'DiscountPct', 'CAST((RAND(CHECKSUM(NEWID()))*(100000)) as smallmoney)'

GO

```

```

CREATE TABLE HumanResources_Employee_MaritalStatus_RandomPseudonymFromSetMap( fromvalue nchar(1),
tovalue nchar(1));
INSERT INTO HumanResources_Employee_MaritalStatus_RandomPseudonymFromSetMap
SELECT t1.fromvalue,t2.tovalue
FROM
(SELECT t.MaritalStatus AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT MaritalStatus FROM HumanResources.Employee) AS t) as t1
INNER JOIN
(SELECT t.MaritalStatus AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT MaritalStatus FROM HumanResources.Employee) AS t) as t2
ON t1.row_num = t2.row_num
GO

CREATE TABLE HumanResources_Shift_ShiftID_RandomPseudonymFromSetMap( fromvalue tinyint, tovalue
tinyint);
INSERT INTO HumanResources_Shift_ShiftID_RandomPseudonymFromSetMap
SELECT t1.fromvalue,t2.tovalue
FROM
(SELECT t.ShiftID AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT ShiftID FROM HumanResources.Shift) AS t) as t1
INNER JOIN
(SELECT t.ShiftID AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT DISTINCT
ShiftID FROM HumanResources.Shift) AS t) as t2
ON t1.row_num = t2.row_num
GO

CREATE TABLE Sales_SalesPerson_BusinessEntityID_RandomPseudonymFromSetMap( fromvalue int, tovalue
int);
INSERT INTO Sales_SalesPerson_BusinessEntityID_RandomPseudonymFromSetMap
SELECT t1.fromvalue,t2.tovalue
FROM
(SELECT t.BusinessEntityID AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM
(SELECT DISTINCT BusinessEntityID FROM Sales.SalesPerson) AS t) as t1
INNER JOIN
(SELECT t.BusinessEntityID AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT BusinessEntityID FROM Sales.SalesPerson) AS t) as t2
ON t1.row_num = t2.row_num

--HumanResources.Employee
GO
IF EXISTS (SELECT column_default FROM vw_TablesColumnsProperties WHERE
schema_table_name='HumanResources.Employee' and column_name='rowguid' and column_default is not
NULL)
AND EXISTS (SELECT column_default FROM vw_TablesColumnsProperties WHERE
schema_table_name='HumanResources.Employee' and column_name='ModifiedDate' and column_default is not
NULL)
BEGIN
exec('CREATE VIEW vw_HumanResources_Employee_Anonymized

```

AS

```

SELECT CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(100000000)) as INT) as BusinessEntityID,
CONVERT(varbinary(5), CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(100000)) as varbinary)) as
OrganizationNode, NULL as OrganizationLevel, CAST(CAST(DATEADD(SECOND, ROUND(((DATEDIFF(SECOND,
'2000-01-01 00:00:00' , '2020-01-01 00:00:00')-1) * RAND(CHECKSUM(NEWID()))), 0), '2000-01-01
00:00:00')) AS datetime) as date) as BirthDate, t1.tovalue as MaritalStatus, t2.tovalue as Gender,
t3.tovalue as SalariedFlag, CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(10000)) as SMALLINT) as
VacationHours, t4.tovalue as SickLeaveHours, CAST(ROUND(RAND(CHECKSUM(NEWID()))),0) AS BIT) as
CurrentFlag, CAST((SELECT CASE WHEN
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')' =
'getdate()' THEN CAST(getdate() as nvarchar(2000))

WHEN
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')' =
'newid()' THEN CAST(newid() as nvarchar(2000))

ELSE
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')'

END as column_default

FROM vw_TablesColumnsProperties WHERE schema_table_name='HumanResources.Employee' and
column_name='rowguid' and column_default is not NULL) as uniqueidentifier) as rowguid,

CAST((SELECT CASE WHEN
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')' =
'getdate()' THEN CAST(getdate() as nvarchar(2000))

WHEN
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')' =
'newid()' THEN CAST(newid() as nvarchar(2000))

ELSE
REPLACE(REPLACE(REPLACE(REPLACE(column_default,'(',')','!'),'(','(',')','')','(',')','!'),'(',')'

END as column_default

FROM vw_TablesColumnsProperties WHERE schema_table_name='HumanResources.Employee' and
column_name='ModifiedDate' and column_default is not NULL) as datetime) as ModifiedDate

FROM HumanResources.Employee t INNER JOIN
HumanResources_Employee_MaritalStatus_RandomPseudonymFromSetMap t1 ON t.[MaritalStatus] =
t1.fromvalue

INNER JOIN (SELECT a.fromvalue, b.tovalue FROM

(SELECT x.Gender AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT Gender FROM HumanResources.Employee) AS x) as a INNER JOIN

(SELECT y.Gender AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT DISTINCT
Gender FROM HumanResources.Employee) AS y) as b ON a.row_num = b.row_num) as t2 ON t.[Gender] =
t2.fromvalue

INNER JOIN bit_RandomPseudonymMap t3 ON t.[SalariedFlag] = t3.fromvalue

INNER JOIN smallint_RandomPseudonymMap t4 ON t.[SickLeaveHours] = t4.fromvalue')

END

ELSE BEGIN print('ERROR WHEN CHECKING DEFAULTS') END

--HumanResources.Shift

GO

CREATE VIEW vw_HumanResources_Shift_Anonymized AS

SELECT t1.tovalue as ShiftID, t2.tovalue as Name, t3.tovalue as StartTime, getdate() as EndTime
FROM HumanResources.Shift t

INNER JOIN HumanResources_Shift_ShiftID_RandomPseudonymFromSetMap t1 ON t.[ShiftID] = t1.fromvalue

INNER JOIN (SELECT a.fromvalue, b.tovalue FROM

```

```

(SELECT x.Name AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT DISTINCT
Name FROM HumanResources.Shift) AS x) as a

INNER JOIN (SELECT y.Name AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT Name FROM HumanResources.Shift) AS y) as b ON a.row_num = b.row_num) as t2

ON t.[Name] = t2.fromvalue

INNER JOIN (SELECT * FROM time_7_RandomPseudonymPerTableMap WHERE columnname='StartTime' and
tablename='HumanResources.Shift') t3 ON t.[StartTime] = t3.fromvalue

--Sales.SpecialOffer

GO

CREATE VIEW vw_Sales_SpecialOffer_Anonymized AS

SELECT CAST(FLOOR(RAND(CHECKSUM(NEWID()))*(1000000000)) as INT) as SpecialOfferID, t1.tovalue as
DiscountPct, t2.tovalue as Category, t3.tovalue as StartDate, t4.tovalue as MinQty, NULL as MaxQty
FROM Sales.SpecialOffer t

INNER JOIN (SELECT * FROM smallmoney_RandomPseudonymPerTableMap WHERE columnname='DiscountPct' and
tablename='Sales.SpecialOffer') t1 ON t.[DiscountPct] = t1.fromvalue

INNER JOIN (SELECT a.fromvalue, b.tovalue FROM

(SELECT x.Category AS fromvalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT Category FROM Sales.SpecialOffer) AS x) as a INNER JOIN

(SELECT y.Category AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM (SELECT
DISTINCT Category FROM Sales.SpecialOffer) AS y) as b ON a.row_num = b.row_num) as t2

ON t.[Category] = t2.fromvalue

INNER JOIN datetime_RandomPseudonymMap t3 ON t.[StartDate] = t3.fromvalue INNER JOIN

(SELECT a.fromvalue, b.tovalue FROM (SELECT x.MinQty AS fromvalue, row_number() over (ORDER BY
NEWID()) AS row_num FROM (SELECT DISTINCT MinQty FROM Sales.SpecialOffer) AS x) as a

INNER JOIN (SELECT y.MinQty AS tovalue, row_number() over (ORDER BY NEWID()) AS row_num FROM
(SELECT DISTINCT MinQty FROM Sales.SpecialOffer) AS y) as b ON a.row_num = b.row_num) as t4

ON t.[MinQty] = t4.fromvalue

```

Drop statements

```

DROP VIEW vw_getRANDValue
DROP FUNCTION GenerateRandomString
DROP PROCEDURE sp_fillup_RandomPseudonymTableMap
DROP PROCEDURE sp_fillup_RandomPseudonymMap
DROP TABLE bit_RandomPseudonymMap
DROP TABLE smallint_RandomPseudonymMap
DROP TABLE datetime_RandomPseudonymMap
DROP TABLE time_7_RandomPseudonymPerTableMap
DROP TABLE smallmoney_RandomPseudonymPerTableMap
DROP TABLE HumanResources_Employee_MaritalStatus_RandomPseudonymFromSetMap
DROP TABLE HumanResources_Shift_ShiftID_RandomPseudonymFromSetMap
DROP TABLE Sales_SalesPerson_BusinessEntityID_RandomPseudonymFromSetMap
DROP VIEW vw_HumanResources_Employee_Anonymized
DROP VIEW vw_HumanResources_Shift_Anonymized
DROP VIEW vw_Sales_SpecialOffer_Anonymized

```