|  |  |
| --- | --- |
| **Soal Praktikum**  *Practicum Case* |  |
| ISYS6028  Database Systems |
| **Teknik Informatika**  *Computer Science* | O221-ISYS6028-DW01-09 |
| **Periode Berlaku Mulai** Semester Ganjil 2017/2018  ***Valid on*** *Odd Semester Year* 2017/2018 | **Revisi 00**  *Revision 00* |

**Learning Outcomes**

* Apply database language and SQL Programming language

## Topic

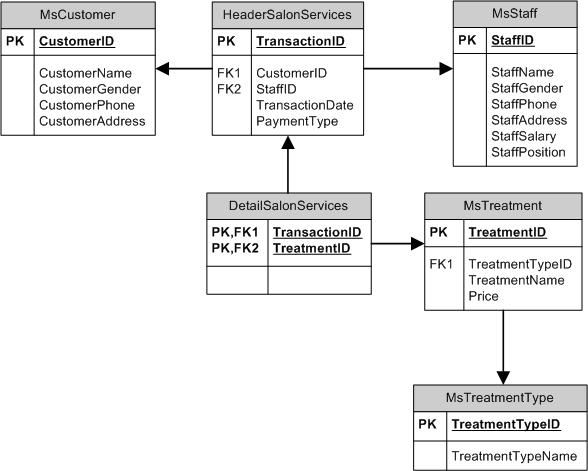
* Session 09 **-** Advanced SQL

## Sub Topics

* Cursors
* Store Procedure
* Trigger
* Function

**Tabel Relasional**

*Relational Table*



**Sintaks**

*Syntax*

**Create Store Procedure**

CREATE {PROC | PROCEDURE} **procedure\_name** [ @parameter datatype ( length ) [, …] ]

AS query

**Alter Store Procedure**

ALTER {PROC | PROCEDURE} **procedure\_name** [ @parameter datatype ( length ) [, …] ]

AS query

Use ‘@’ for variable name, e.g.: @customerName

**Execute Store Procedure**

[EXEC | EXECUTE] **procedure\_name** [ parameter [, …] ]

**Drop Store Procedure**

DROP {PROC | PROCEDURE} **procedure\_ name**

**Create Trigger**

CREATE TRIGGER **trigger\_name** ON {table\_name | view\_name}

{ FOR | AFTER | INSTEAD OF } { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }

AS query

After trigger not supported for views

**Drop Trigger**

DROP TRIGGER **trigger\_name**

**Cursor**

DECLARE **cursor\_name** CURSOR

[LOCAL | GLOBAL]

[FORWARD\_ONLY | SCROLL]

[STATIC | KEYSET | DYNAMIC | FAST\_FORWARD]

[READ\_ONLY | SCROLL\_LOCKS | OPTIMISTIC]

[TYPE\_WARNING]

FOR select\_query

OPEN **cursor\_name**

FETCH [ NEXT | PRIOR | FIRST | LAST

| ABSOLUTE { row\_number }

| RELATIVE { row\_number } ]

FROM [GLOBAL] **cursor\_name**

WHILE @@FETCH\_STATUS = 0

BEGIN

query

FETCH NEXT FROM **cursor\_name**

END

CLOSE [GLOBAL] **cursor\_name**

DEALLOCATE [GLOBAL] **cursor\_name**

## Soal

*Case*

1. Create a store procedure with named ‘sp1’ to display CustomerId, CustomerName, CustomerGender, and CustomerAddress for every Customer with Id based on user’s input.

(**create procedure**)

|  |
| --- |
| EXEC sp1 ‘CU001’ |
| D:\08\1.png |

1. Create a store procedure with named ‘sp2’ that receives CustomerName as input from user with the following specification:

* If the length of CustomerName is odd then procedure will give output ‘**Character Length of Mentor Name is an Odd Number’**.
* If the length of CustomerName is even then procedure will display CustomerId, CustomerName, CustomerGender, TransactionId, and TransactionDate for every transaction with customer whose name contains the name that was inputted by user.

(**create procedure**, **len**, **like**)

|  |
| --- |
| EXEC sp2 ‘Elysia Chen’ |
| D:\08\2a.png |

|  |
| --- |
| EXEC sp2 ‘Fran’ |
| D:\08\2b.png |

1. Create a store procedure named ‘sp3’ to update StaffId, StaffName, StaffGender, and StaffPhone on **MsStaff** table based on StaffId, StaffName, StaffGender, and StaffPhone that was inputted by user. Then display the updated data if the StaffId exists in **MsStaff** table. Otherwise show message ‘**Staff does not exists**’.

(**create procedure**, **update**, **exists**)

|  |
| --- |
| Before EXEC sp3 'SF005', 'Ryan Nixon', 'M', '08567756123' |
| D:\08\3before.png |

|  |
| --- |
| After EXEC sp3 'SF005', 'Ryan Nixon', 'M', '08567756123' |
| D:\08\3a.png |

|  |
| --- |
| EXEC sp3 'SF008', 'Ryan Nixon', 'M', '08567756123' |
| D:\08\3b.png |

1. Create trigger named ‘trig1’ for **MsCustomer** table to validate if there are any data which had been updated, it will display before and after updated data on **MsCustomer** table.

(**create trigger**, **union**)

|  |
| --- |
| Update MsCustomer SET CustomerName = 'Franky Quo' WHERE CustomerId = 'CU001' |
| D:\08\4.png |

1. Create trigger with name ‘trig2’ for **MsCustomer** table to validate if there are any new inserted data, then the first data on **MsCustomer** will be deleted.

(**create trigger**, **top**, **delete**)

|  |
| --- |
| Before INSERT INTO MsCustomer VALUES('CU006','Yogie soesanto', 'Male', '085562133000', 'Pelsakih Street no 52') |
| D:\08\5before.png |

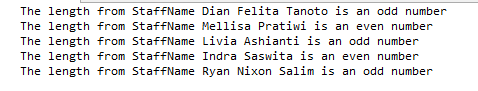
|  |
| --- |
| After INSERT INTO MsCustomer VALUES('CU006','Yogie soesanto', 'Male', '085562133000', 'Pelsakih Street no 52') |
| D:\08\5after.png |

1. Create trigger with name ‘trig3’ on **MsCustomer** table to validate if the data on **MsCustomer** table is deleted, then the deleted data will be insert into **Removed** table. If **Removed** table hasn’t been created, then create the **Removed** table and insert the deleted data to **Removed** table.

(**create trigger**, **object\_id**, **is not null**, **insert**, **select into**)

|  |
| --- |
| DELETE FROM MsCustomer WHERE CustomerId = 'CU002' |
| D:\08\6.png |

1. Create cursor with name ‘cur1’ to validate whether the length of StaffName is odd or even then show the message about result.

(**declare cursor**, **len**)

1. Create procedure named ‘sp4’ that receive StaffName from user’s input to display StaffName and StaffPosition for every staff which name contains the word that has been inputted by user.

(**create procedure**, **declare cursor**, **like**)

|  |
| --- |
| EXEC sp4 'a' |
| D:\08\8.png |

1. Create procedure with name ‘sp5’ that receive CustomerId from user’s input to display CustomerName, and TransactionDate for every customer which Id has been inputted by user and did treatment which ID is an even number.

(**create procedure**, **declare cursor**, **in**, **right**)

|  |
| --- |
| EXEC sp5 'CU001' |
| D:\08\9.png |

1. Delete all procedure and trigger that has been made.

(**drop proc**, **drop trigger**)