

Version: 03.01

Date: 25-Mar-2016

Developed by:

Verified by:

Endava SQL Discipline

SQL Test

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date of revision** | **Description of modifications** |
| 03.01 | 25-March-2016 | The third version of the document. |

# Description

**About the test:**

|  |  |
| --- | --- |
|  |  |
| Applied Level | Intermediate, Advanced |
| Number of tasks | 7 |
| Domain | Standard DB |
| Test type | PC |
| Test duration | 2 h |

**Evaluation info:**

|  |  |
| --- | --- |
|  |  |
| Evaluated person |  |
| Evaluator name |  |
| Date of evaluation |  |
| Evaluation result  (passed/failed) |  |

# Test Tasks

## Precondition:

Use the below credentials:

|  |  |
| --- | --- |
| **Server Name** | MDCH-AMWCI-S01 |
|
| **DB** | AMInternship |
|
|

## Task 1

**Create a new table “History\_Person” with the columns: id, action\*, change date, id person and all the “Person” columns with the new and old values (i.e. first\_name\_old, first\_name\_new, ….) .**

**\*The action column should store the DML operation (INSERT, UPDATE or DELETE).**

**Query:**

|  |
| --- |
| create table History\_Person (  id int,  action varchar(15),  change date,  id\_person int,  first\_name\_old varchar(20),  first\_name\_new varchar(20),  last\_name\_old varchar(20),  last\_name\_new varchar(20),  date\_old date,  date\_new date,  gender\_id\_old int,  gender\_id\_new int,  address\_id\_old int,  address\_id\_new int,  job\_id\_old int,  job\_id\_new int,  title\_id\_old int,  title\_id\_new int  ) |

## Task 2

**Create 3 different triggers for INSERT, UPDATE and DELETE.**

**Query:**

|  |
| --- |
| /\*Insert trigger\*/  CREATE TRIGGER Insert\_Trigger\_On\_Person  ON Person  AFTER INSERT  AS  Declare @id INT,@action varchar(10), @change date,@id\_person int,@first\_name varchar(15), @last\_name varchar(15),@date\_person date,  @gender\_id int, @address\_id int, @job\_id int, @title\_id int  select  @id= p.id,  @first\_name=p.first\_name,  @last\_name=p.last\_name,  @date\_person=p.date,  @gender\_id=p.gender\_id,  @address\_id=p.address\_id,  @job\_id=p.job\_id,  @title\_id=p.title\_id  from Person p  SET @action ='Insert'  SET @change = getdate()  BEGIN  INSERT INTO History\_Person  (id\_person,action,change,first\_name\_old,last\_name\_old,date\_old,gender\_id\_old,address\_id\_old,job\_id\_old,title\_id\_old)  values(@id,@action,@change,@first\_name,@last\_name,@date\_person,@gender\_id,@address\_id,@job\_id,@title\_id)  END  GO  /\*Delete trigger\*/  CREATE TRIGGER Delete\_Trigger\_On\_Person  ON Person  AFTER DELETE  AS  Declare @id INT,@action varchar(10), @change date,@id\_person int,@first\_name varchar(15), @last\_name varchar(15),@date\_person date,  @gender\_id int, @address\_id int, @job\_id int, @title\_id int  select  @id\_person= p.id,  @first\_name=p.first\_name,  @last\_name=p.last\_name,  @date\_person=p.date,  @gender\_id=p.gender\_id,  @address\_id=p.address\_id,  @job\_id=p.job\_id,  @title\_id=p.title\_id  from Person p  BEGIN  SET @action ='Delete'  SET @change = getdate()  INSERT INTO History\_Person  (id\_person,action,change,first\_name\_old,last\_name\_old,date\_old,gender\_id\_old,address\_id\_old,job\_id\_old,title\_id\_old)  values(@id\_person,@action,@change,@first\_name,@last\_name,@date\_person,@gender\_id,@address\_id,@job\_id,@title\_id)  END  GO    /\*Update trigger\*/  CREATE TRIGGER Update\_Trigger\_On\_Person  ON Person  AFTER UPDATE  AS  Declare @id INT,@action varchar(10), @change date,@id\_person int,@first\_name varchar(15), @last\_name varchar(15),@date\_person date,  @gender\_id int, @address\_id int, @job\_id int, @title\_id int  select  @id\_person= p.id,  @first\_name=p.first\_name,  @last\_name=p.last\_name,  @date\_person=p.date,  @gender\_id=p.gender\_id,  @address\_id=p.address\_id,  @job\_id=p.job\_id,  @title\_id=p.title\_id  from Person p  BEGIN    SET @action ='UPDATE'  SET @change = getdate()  UPDATE History\_Person  SET  action=@action,  change=@change,  first\_name\_new=@first\_name,  last\_name\_new=@last\_name,  date\_new=@date\_person,  gender\_id\_new=@gender\_id,  address\_id\_new=@address\_id,  job\_id\_new=@job\_id,  title\_id\_new=@title\_id  WHERE id\_person=@id\_person  END  GO |

## Task 3

**Insert a new record in the “Person table” with the values: “John”, “Smith”, “1988-06-01”, “2”, “1”, “1”, “2”.**

**Update the entered record with a new information changing only the last name and birth date.**

**Delete ONLY the updated record. Check the “History\_Person” table.**

**Query:**

|  |
| --- |
| insert into person (first\_name,last\_name,date, gender\_id,address\_id,job\_id,id) values ('John','Smith','1988-06-01',2,1,1,4)  update person  set last\_name='Cruz',  date='1998-11-01'  where first\_name='John'  delete from person where last\_name='Cruz' |

## Task 4

**Create a new table “History\_Salary” with the columns: id, change date, person id, job id, salary old, salary new and salary diff.**

**Query:**

|  |
| --- |
| create table History\_Salary(  ID int IDENTITY(1,1) ,  change date,  person\_id int,  job\_id int,  salary\_old int,  salary\_new int,  salary\_diff int  ) |

## Task 5

**Create a trigger which will store the salary changes in the “History\_Salary” table.**

**Update the salary for the entered record at the task 21.**

**Check the “History\_Salary” table.**

**Query:**

|  |
| --- |
| CREATE TRIGGER Update\_Trigger\_On\_Salary  ON salary  AFTER UPDATE  AS  Declare @id INT, @change date,@person\_id int, @job\_id int, @salary\_old int,@salary\_new int, @salary\_diff int  select  @person\_id=p.id,  @job\_id=j.id,  @salary\_old=s.salary,  @salary\_new=s.salary  from person p  inner join job j On j.id=p.job\_id  inner join salary s On s.id=j.salary\_id  BEGIN  SET @change = getdate()  INSERT INTO History\_Salary  (change,person\_id,job\_id,salary\_old,salary\_new,salary\_diff)  values(@change,@person\_id,@job\_id,@salary\_old,@salary\_new,@salary\_old-@salary\_new)  END |

## Task 6

**Create a new table “Accountancy” with the columns: id, name, salary.**

**Query:**

|  |
| --- |
| create table Accountancy (  id int,  name varchar(15),  salary int  ) |

## Task 7

**Create a stored procedure using a CURSOR which will populate the “Accountancy” table with the values in the order: id, name (first name and last name), salary.**

**Check the “Accountancy” table.**

**Query:**

|  |
| --- |
| CREATE PROCEDURE my\_proc  AS  DECLARE  @id AS INT,  @name AS varchar(15),  @salary AS int  Declare curP cursor For  select  p.id,  p.first\_name,  s.salary  from  person p  inner join job j On j.id=p.job\_id  inner join salary s On s.id=j.salary\_id  OPEN curP  Fetch Next From curP Into @id, @name,@salary  While @@Fetch\_Status = 0 Begin  insert into Accountancy (id,name,salary) values (@id,@name,@salary)    Fetch Next From curP Into @id, @name,@salary  End -- End of Fetch  Close curP  Deallocate curP  exec my\_proc  select \* from Accountancy |