**IMPORTANT NOTES**

**1)EVENT BUBBLING**

<thng-click="getSortColumn('RequestId')"> Request ID<divng-class="getSortClass('RequestId')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.RequestId"/></th>

HERE CLICK EVENT WILL WORK ONLY ON TH AND DIV BUT NOT ON INPUT TAG BECAUSE WE HAVE WRITTEN onclick=event.stopPropogation(). This is called preventing **EVENT BUBBLING**

**2)FILTER AND SEARCH OPTIONS IN TABLE USING ANGULAR JS**

<tableclass="table report-result">

<theadstyle="background:hsl(69deg100%61%);">

<tr>

</tr>

<tr>

<thng-click="getSortColumn('RequestId')">

Request ID<divdata-ng-class="getSortClass('RequestId')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.RequestId"/>

</th>

<thng-click="getSortColumn('RequestedBy')">

Resource Owner<divng-class="getSortClass('RequestedBy')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.RequestedBy"/></th>

<thng-click="getSortColumn('ResourceType')">

Resource Type<divng-class="getSortClass('ResourceType')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.ResourceType"/></th>

<thng-click="getSortColumn('CandidateName')">

Candidate Name<divng-class="getSortClass('CandidateName')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.CandidateName"/></th>

<thng-click="getSortColumn('Title')">

Technology<divng-class="getSortClass('Title')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.Title"/></th>

<thng-click="getSortColumn('JoiningDate')">

Joining Date<divng-class="getSortClass('JoiningDate')"></div>

<inputonclick="event.stopPropagation()"type="text"ng-model="searchColumn.JoiningDate"/></th>

</tr>

</thead>

<tbody>

<trng-repeat=" row**in**dataList|**orderBy**:sortColumn:reverseSort|**filter**:searchColumn">

<tdstyle="width:100px"align="center">**{{**row.RequestId**}}**</td>

<tdstyle="width:100px"align="center">**{{**row.RequestedBy**}}**</td>

<tdstyle="width:100px"align="center">**{{**row.ResourceType**}}**</td>

<tdstyle="width:100px"align="center">**{{**row.CandidateName**}}**</td>

<tdstyle="width:100px"align="center">**{{**row.Title**}}**</td>

<tdstyle="width:100px"align="center">**{{**row.JoiningDate**}}**</td>

</tr>

</tbody>

</table>

FUNCTIONS ARE:-

$scope.getSortClass = function (column) {

if ($scope.sortColumn == column) {

return $scope.reverseSort ? 'arrow-down' : 'arrow-up';

} elsereturn'';

}

$scope.getSortColumn = function (column) {

$scope.reverseSort = ($scope.sortColumn == column) ? !$scope.reverseSort : false;

$scope.sortColumn = column;

}

**3)DATE FORMAT FUNCTONS FOR .CS:-**

publicstringGetExportPMO(stringStartDate, stringEndDate)

{

System.Globalization.CultureInfo culture = newSystem.Globalization.CultureInfo("en-us");

StartDate = Convert.ToDateTime(StartDate).ToString("yyyy-MM-dd", culture);

EndDate = Convert.ToDateTime(EndDate).ToString("yyyy-MM-dd", culture);

try

{

DALResourceoResorce = newDALResource();

stringExcelUrl = oResorce.DALGetExportPMO(StartDate, EndDate);

returnExcelUrl;

}

catch (Exception ex)

{

throw ex;

}

}

**4)CALENDAR FOR HTML PAGE**

<divclass="col-sm-3">

<divclass="form-group col-md-6 multi-datepicker form-inline">

<label>Start Date<spanstyle="color: red;font-size: xx-small">\*</span></label>

<inputreadonly="readonly"id="txtDateFrom"ng-model="Startdate"max="txtDateFrom"class="form-control"datepicker-popup="dd-MMM-yy"close-on-date-selection='false'requiredng-click='opened=true'is-open='opened'style="width: 100%; border:0px!important; border-bottom:1pxsolid#ddd!important; border-radius:0px!important;"/>

</div>

</div>

<divclass="col-sm-3">

<divclass="form-group col-md-6 multi-datepicker form-inline">

<label>End Date<spanstyle="color: red;font-size: xx-small">\*</span></label>

<inputreadonly="readonly"id="txtDateTo"ng-model="Enddate"max="txtDateTo"class="form-control"datepicker-popup="dd-MMM-yy"close-on-date-selection='false'requiredng-click='opened1=true'is-open='opened1'style="width: 100%; border:0px!important; border-bottom:1pxsolid#ddd!important; border-radius:0px!important;"/>

</div>

</div>

**4)Function in SQL to split strings by delimeter and store it in table**

ALTERFUNCTION [dbo].[StringSplit]

(

@String VARCHAR(MAX), @Separator CHAR(1)

)

RETURNS @RESULT TABLE(ValueVARCHAR(MAX))

AS

BEGIN

DECLARE @SeparatorPositionINT=CHARINDEX(@Separator, @String ),

@Value VARCHAR(MAX), @StartPositionINT= 1

IF @SeparatorPosition= 0

BEGIN

INSERTINTO @RESULT VALUES(@String)

RETURN

END

SET @String = @String + @Separator

WHILE @SeparatorPosition> 0

BEGIN

SET @Value =SUBSTRING(@String , @StartPosition, @SeparatorPosition- @StartPosition)

IF( @Value <>'')

INSERTINTO @RESULT VALUES(@Value)

SET @StartPosition= @SeparatorPosition+ 1

SET @SeparatorPosition=CHARINDEX(@Separator, @String , @StartPosition)

END

RETURN

END

**Execution:**

**select \* from stringsplit(‘aquib,javed’,’,’)**

**5)Last date of the current month**

DATEADD(s,-1,DATEADD(mm,DATEDIFF(m,0,GETDATE())+1,0))

**6)When setting identity\_insert on two tables then if one is set off then only other can be set to on**

SETIDENTITY\_INSERT [BetaHiring\_Pro].[dbo].[CandidateDetailOld] OFF;

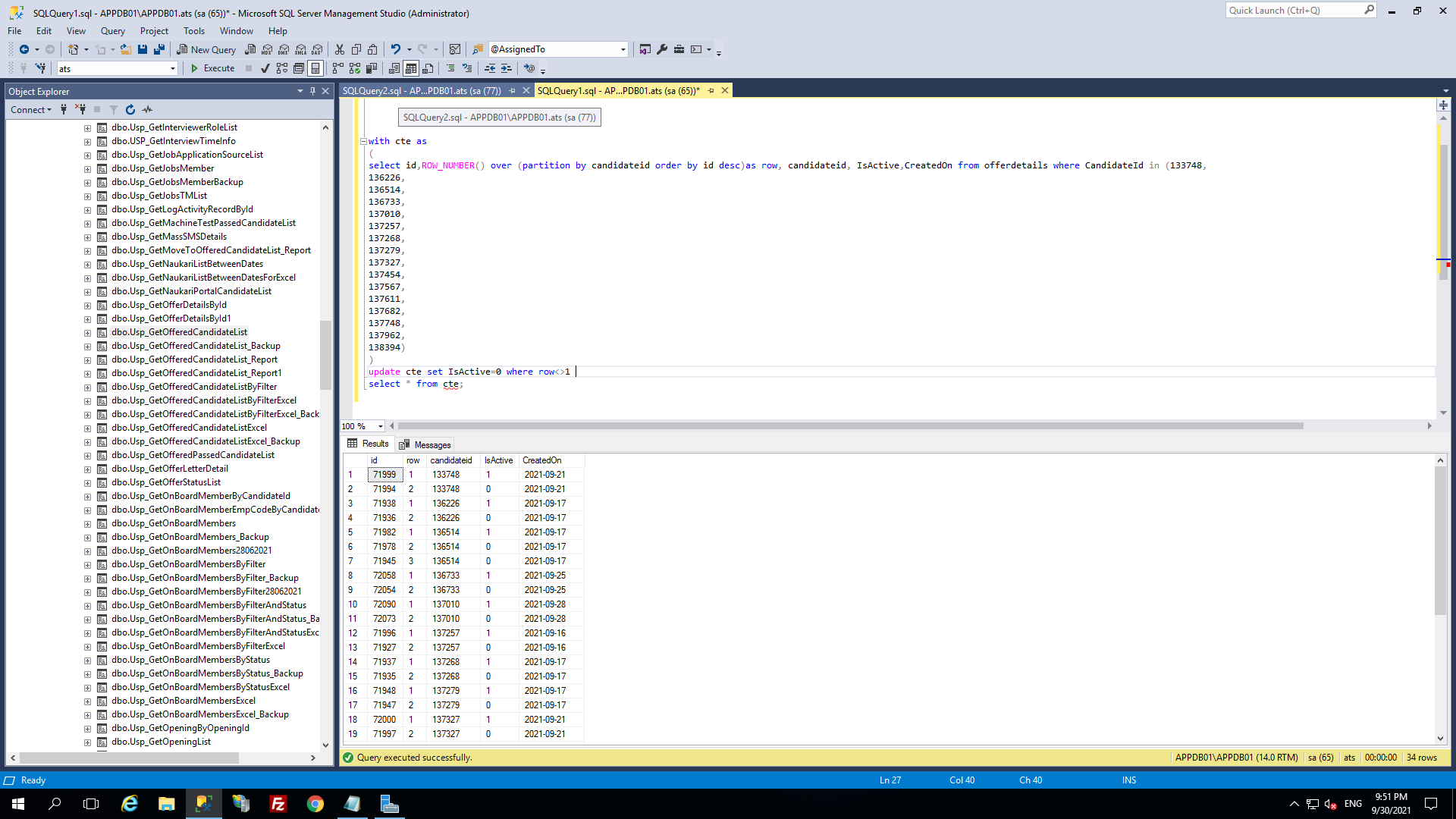
SETIDENTITY\_INSERT [atsbeta].[dbo].[CandidateOld] ON;

**7)To create a table with same structure as existing one**

Select \* into <new table>from <table name> where 1=2

**8)<>**

**9)removes duplicate records by keeping latest one**



**10)Removes extra spaces from string in sql**

replace(replace(replace(B.FName+' '+ISNULL(B.MName,'')+' '+ISNULL(B.LName,'')+ISNULL(B.NameSuffix,''),' ','<>'),'><',''),'<>',' ')asName,

**11)Loaders**

<https://bootsnipp.com/snippets/GaeVl>

**use this code to run loader in foreground(put loader wrap class in loader div:-**

.loader\_wrap {

    width: 100%;

height: 100%;

position: fixed;

left: 0;

top: 0;

justify-content: center;

align-items: center;

display: flex;

background: #00000063;

flex-direction: column;

z-index: 9999;

}

.loader\_wrap.loadp {

    color: #fff;

}

**12)Dialog box for angular:**

**Html:**

<divclass="backdrop"[ngStyle]="{'display':display}"></div>

<!-- modal -->

<divclass="modal"tabindex="-1"role="dialog"[ngStyle]="{'display':display}">

    <!-- modal-dialog -->

    <divclass="modal-dialog modal-dialog-centered"role="document">

        <!-- modal-content -->

        <divclass="modal-content">

            <!-- modal-header -->

            <divclass="modal-header">

                <h4class="modal-title">Export File</h4>

                <buttontype="button"class="close"aria-label="Close"(click)="closeModalDialog()"><spanaria-hidden="true">&times;</span></button>

                <!--<h4 class="modal-title">Login</h4> -->

            </div>

            <!-- modal-body -->

            <divclass="modal-body">

                <divclass="d-flex justify-content-between mb-3">

                    <pclass="mb-0">

                      First Name:<inputtype="text"[(ngModel)]="first\_name"name="first\_name"><br>

                      Last Name:<inputtype="text"[(ngModel)]="last\_name"name="last\_name"><br>

                      Contact No:<inputtype="text"[(ngModel)]="phone"name="phone">

                    </p>

                </div>

            </div>

            <!-- modal-footer -->

            <divclass="modal-footer">

              <buttontype="button"class="btnbtn-default"(click)="saveDtls()">Save</button>

                <buttontype="button"class="btnbtn-default"(click)="closeModalDialog()">Close</button>

            </div>

        </div>

    </div>

</div>

**css**

.modal.modal-header {

    display: block;

}

.modal.modal-header.close {

    position: relative;

    top: -10px;

    opacity: .5;

}

.modal {

    background: #00000080;

}

.modal-header{

    background-color: #0096a3;

}

.modal-header.close {

    border: 0;

    background: transparent;

    padding: 0;

    margin: 0;

    line-height: 0;

}

.modal-headerh4 {

    color: #fff;

}

.modal-header.closespan {

    font-size: 30px;

    color: #fff;

}

.modal-footer.btn {

    background: #df040421;

    color: red;

    border-color: transparent;

}

**.ts:**

closeModalDialog(){

this.display='none';

}

**12)Profiler in SQL Server:**

**Tools>>SQL Server Profiler in SSMS will display which procedure was called and when**

**13)Searches text in all stored procedure:**

SELECTDISTINCT

o.nameASObject\_Name,

o.type\_desc

FROMsys.sql\_modules m

INNERJOIN

sys.objects o

ONm.object\_id=o.object\_id

WHEREm.definitionLike'%delete from candidate %';

**14)Generating error logs in sql server:**

USE [atsbeta]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[UspATSErrorLog] Script Date: 16-05-2022 20:40:20 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[UspATSErrorLog]

AS

BEGIN

SET NOCOUNT ON

INSERT INTO [dbo].[ATSErrorLog]

(

ErrorNumber

,ErrorDescription

,ErrorProcedure

,ErrorState

,ErrorSeverity

,ErrorLine

,ErrorTime

)

VALUES

(

isnull(ERROR\_NUMBER(),-99)

,ERROR\_MESSAGE()

,ERROR\_PROCEDURE()

,ERROR\_STATE()

,ERROR\_SEVERITY()

,ERROR\_LINE()

,GETDATE()

);

SET NOCOUNT OFF

END

Note:Execute this procedure in the catch block

**MongoDB**

1)View all database-show dbs

2)create a new or switch database-use <dbName>

3)view current database- db

4)Drop database-db.dropDatabase()

5)Create a collection-db.createCollection(<collectionName>)

6)display collections-show collections

7)delete a collection- db.<collectionName>.drop()

8)AGGREGATION is applied in three ways-aggregation pipeline,map-reduce function,single purpose aggregation methods

**angular**

1. **To use bootstrap in angular:**

-npm install bootstrap --save

-@import'bootstrap/dist/css/bootstrap.min.css'; (do this in css file)-sometimes it is required

-updateangular.json under style section add-'node\_modules/bootstrap/dist/css/bootstrap.min.css'

1. **To disable initialization in angular just add to tsconfig.json below strict:true:**

 "strict": true,

    "strictPropertyInitialization": false,

1. **After updating node do the following if facing issue:**

* I deleted my C:/Users/{username}/AppData/Roaming/npm and C:/Users/{username}/AppData/Roaming/npm-cache and reinstalled global npm modules, and the problem ceased.
* It would be much better though if these steps could be avoided after upgrading/downgrading node. If not, then at least, make it explicit in the docs that this step may be necessary when upgrading/downgrading existing node installations.

**4)When this error occurs:-**

**ng : File C:\Users\aquibj\AppData\Roaming\npm\ng.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about\_Execution\_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.**

Solution) run this command in project folder-

set-ExecutionPolicy RemoteSigned -Scope CurrentUser