**course table stored procedures**

SelectCourse @Crs\_Id int = NULL

-Take the course id and return all data of this course if present

course\_Insert

(

@Crs\_Id INTEGER = NULL,

[@Crs\_Name VARCHAR(20) = NULL],

[@Crs\_Duration INTEGER = NULL],

[@Top\_Id INTEGER = NULL]

)

* Used course\_Insert Data to the course table
* Take arguments @id [Crs\_Id]
* @Crs\_Name [Crs\_Name text]
* @Crs\_Duration [Crs\_Duration Time]
* The @Crs\_Id must be unique and the @Top\_Id must be present in the Topic table
* may give only the @Crs\_Id and the rest will be inserted as null

course\_Update

(

@Crs\_Id INTEGER = NULL,

[@Crs\_Name VARCHAR(20) = NULL],

[@Crs\_Duration INTEGER = NULL],

[@Top\_Id INTEGER = NULL]

)

* used to course\_Update inside the course table by the course id
* Take arguments @Crs\_Id [the course id]
* @Crs\_Name [Crs\_Name text]
* @Crs\_Duration [Crs\_Duration Time]
* @Top\_Id [the topic id that this course is for]
* The @Crs\_Id must be unique
* the @Top\_Id must be present in the Topic table
* Can update one or more than one column

course\_Delete @Crs\_Id int = NULL

* take the course id and delete the course Data from the course table if it is present

**Stud\_Course table stored procedures**

Stud\_Course\_Select @Crs\_Id int = NULL, @St\_Id int = NULL

-Take the course id and Student id then return all data of this Stud\_Course if present

Stud\_Courses\_Insert

(

@Crs\_Id INTEGER = NULL,

@St\_Id INTEGER = NULL,

@Grade INTEGER = NULL

)

* Used Stud\_Courses\_Insert Data to the Stud\_Course table
* Take arguments @id1 [Crs\_Id] and @id2[St\_Id] As [composite key]
* @Grade [Crs\_Stud\_Grade]
* @Crs\_Id and @St\_Id As [composite key] must be unique
* @Crs\_Id must be present in the Course table
* @St\_Id must be present in the Student table
* may give only the @Crs\_Id and @St\_Id As [composite key] and @Grade will be inserted as null

Stud\_Course\_Update

(

@Crs\_Id INTEGER = NULL,

@St\_Id INTEGER = NULL,

@Grade INTEGER = NULL

)

* used to Stud\_Course\_Update inside the Stud\_Course table

by the course id and the Student id [Composite key]

* Take arguments @Crs\_Id [the course id] and @St\_Id [the Student id] As [composite key]
* @Grade [Crs\_Stud\_Grade]
* The @Crs\_Id and @St\_Id [Composite key] must be unique
* @Crs\_Id must be present in the Course table
* @St\_Id must be present in the Student table
* Can update one or more than one column

Stud\_Course\_Delete @Crs\_Id int = NULL, @St\_Id int = NULL

* take the course id and Student id As [Composite key] then delete the Student Course Data from the Stud\_Course table if it is present

**ExamAnswer stored procedure**

(

@Ex\_Id int,

@Std\_Id int

)

* used to return St\_Answer from Student\_Exam\_Answer table
* takes @Ex\_Id as the Exam id and @Std\_Id as the Student id [composite key]

**courseName\_numStudent\_perCourse stored procedure**

(

@Ins\_Id int

)

* takes @Ins\_Id as the Instructor id and must exist in Instructor table
* used to return Crs\_Name and numStudent\_perCourse
* numStudent\_perCourse by using this Function count () and giving it St\_Id as parameter
* doing join between tables Student S, Stud\_Course SC, Course C, Ins \_Course IC, Instructor I where S. St\_Id =SC.St\_Id and SC. Crs\_Id=C.Crs\_Id and C. Crs\_Id =IC.Crs\_Id

and IC. Ins\_Id=I.Ins\_Id

* group by (C. Crs\_Name)