**SSRS Reports Documentation**

**1.Student Information Report**

Repot name: Student Details by Track.

This report provides detailed information about students in a specified track.

* 1. **Data source and datasets**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: GetStudentsbyDepartment

Query Description: Returns Student IDs, Students full-name, Student mail, Student Gender, Student Faculty, Student Faculty Grade, Student ITI Status, and Department name.

Parameters: @Track\_ID

* 1. **Report Layout**

Page is consistent of a header, a table, and a footer.

Header content: ITI Logo, Title, a line, and a Sub-title.

Table content: Student ID, Full-name, Gender, Address, marital status, faculty, birthdate, faculty grade, and ITI status.

Footer content: Page number.

Header Title Expression: ="Student Details In " & First(Fields!Track\_Name.Value, "DataSet1")

Sub-title Expression: ="Department Name: " & First(Fields!Department\_Name.Value, "DataSet1")

1. **Student Grades Report**

Repot name: Student Grades.

This report show all specified student grades across all courses.

* 1. **Data source and datasets**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: GetGradebyStudent

Query Description: Returns Student ID, Student Full-Name, Exam ID, Course Name, Total Grade.

Parameters: @Student\_ID

* 1. **Report Layout**

Page is consistent of a header, and a table.

Header content: ITI Logo, Title, a line, and a Sub-title.

Table content: Course name, Exam ID, Total Grade, and an indicator.

Header title Expression: =First(Fields!StudentFullname.Value, "DataSet1") & "'s Grades"

Indicator Properties:

Red: <59

Yellow: <84

Green: >85

1. **Topics in Courses Report**

Report Name: Course Topics

This report shows every topic in a specified course.

* 1. **Data source and datasets**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: Get\_Course\_Topics

Query Description: Returns Topic ID and Topic Name.

Parameters: @CourseID

* 1. **Report Layout**

Page is consistent of a header, and a table.

Header content: ITI Logo, Title, a line, and a Sub-title.

Table content: Topic ID and Topic Name.

Header title: Report Name Box.

Sub-title Expression: ="Course: "& Parameters!CourseID.Label.

1. **Instructor Courses Report**

Report name: Instructor courses and number of students

This report shows the courses taught by the specified instructor and the number of students taught by the specified instructor.

* 1. **Data source and datasets**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: Get\_Ins\_Students

Query Description: Returns Insturctor full-name, Course ID, course name, and number of students

Parameters: @insID

* 1. **Report Layout**

Page is consistent of a header, and a table.

Header content: ITI Logo, Title, and a line.

Table content: Instructor name, Course ID, Course name, and Number of Students.

Header title: Report Name Box.

1. **Exam Questions Report**

Report Name: Exam Questions.

This report shows the questions and their answers in a specified exam.

* 1. **Data source and datasets**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: GetExamQuestionsWithChoices

Parameter: @ExamNumber

* 1. **Report Layout**

Page is consistent of a header, and a List.

Header content: ITI Logo, Title, a line, and sub-title.

List content: Column header, Question ID, Question Description, and Choices list.

Header Expression: =First(Fields!Course\_Name.Value, "DataSet1") & " Exam"

Sub-title Expression: ="Exam ID: " & Parameters!ExamNumber.Value

Choices list Expression: =Join(

LookupSet(

Fields!Question\_ID.Value,

Fields!Question\_ID.Value,

IIF(

Fields!Choice\_Text.Value Is Nothing OR Fields!Choice\_Text.Value = "",

" True/False",

" " & Fields!ChoiceNumber.Value & ". " & Fields!Choice\_Text.Value

),

"DataSet2"

),

vbCrLf

)

1. **Student Answers Report**

Report name: Student answers in exam.

This report shows a specified student answers in a specified exam.

* 1. **Data Source and dataset**

Data source: ITIExaminationsystem database.

Data Type: Stored Procedure.

Query name: GetStudentExamAnswers

Query Description: Returns Question ID, Question Description, Question Correct Answer, Student Answer, Exam ID, Student ID, Student Full name, and Course name.

Parameter: @Exam\_ID & @Student\_ID

* 1. **Report Layout**

Page is consistent of a header, and a Table.

Header content: ITI Logo, Title, a line, and sub-title.

Table content: Question ID, Question, Student Answer, Correct Answer, and a correct sign as an Image.

Title Expression: ="Student Answers in" & vbCrLf &

"""" & First(Fields!course\_name.Value, "DataSet1") & """" & vbCrLf &

"Exam ID: " & Parameters!Exam\_ID.Value

Sub-tittle Expression: ="Student name:" & First(Fields!StudentName.Value, "DataSet1")

Image Column Visibility Expression: =IIf(Fields!Question\_Model\_Answer.Value=Fields!Student\_Answer.Value, False, True)s

1. **Stored Procedures:** 
   1. Report 1:

**GetStudentsbyDepartment**  
create or alter procedure getstudentsbydepartment @track\_ID int

as

begin

select Student\_ID,Track\_Name ,Student\_Fname+' '+Student\_Lname as StudentFullname, Student\_Mail,Student\_Gender,

Student\_Address, Student\_Birthdate,Student\_Marital\_Status,

Student\_Faculty, Student\_Faculty\_Grade,Student\_ITI\_Status,d.Department\_Name

from Student s join [Group] as g

on s.Intake\_Branch\_Track\_ID = g.group\_id join Track t

on t.Track\_ID = g.track\_id join Department d

on t.Department\_ID = d.Department\_ID

where t.Track\_ID = @track\_ID

end

* 1. Report 2:

**GetGradebyStudent**

Create procedure [dbo].[getgradebystudent] @student\_id int

as

begin

select sea.student\_id, s.Student\_Fname+' '+s.Student\_Lname as StudentFullname, c.course\_name as [Course Name], sea.exam\_id,Sum(sea.Student\_grade)\*100 / count(sea.question\_id) as [Total Grade]

from

student s join

student\_exam\_answer as sea on s.student\_id = sea.student\_id

join exam e on e.exam\_id = sea.exam\_id

join course as c on e.course\_id = c.course\_id

where sea.student\_id = @student\_id

group by sea.student\_id, sea.exam\_id, s.student\_fname, s.student\_lname, c.course\_name

end

* 1. Report 3:

**Get\_Course\_Topics**

create proc Get\_Course\_Topics @CourseID int

as

select t.Topic\_ID, t.Topic\_Name from Topic t inner join course c

on t.Course\_ID = c.Course\_ID

where t.Course\_ID = @CourseID

* 1. Report 4:

**Get\_Ins\_Students**

create or alter proc Get\_Ins\_Students @insID int

as

select concat(ins.Instructor\_fname,' ', ins.Instructor\_Lname) as InsFullName, insc.course\_ID, c.Course\_name, count(sc.student\_ID) as NumberOfStudents

from Instructor ins

inner join Instructor\_Course insc on ins.Instructor\_ID = insc.Instructor\_ID

inner join Student\_Course sc on sc.Course\_ID = insc.Course\_ID

inner join course c on c.Course\_ID = insc.Course\_ID

where ins.instructor\_ID = @insID

group by ins.Instructor\_fname, ins.Instructor\_Lname, insc.course\_ID, c.Course\_Name

* 1. Report 5:

**GetExamQuestionsWithChoices**

Create PROCEDURE [dbo].[GetExamQuestionsWithChoices]

@ExamNumber INT

AS

BEGIN

SET NOCOUNT ON;

-- First result set: Questions only (no duplicates)

SELECT DISTINCT

eq.Exam\_ID,

qb.Question\_ID,

qb.Question\_Type,

qb.Question\_Description,

qb.Question\_Model\_Answer,

c.Course\_Name

FROM [ITIExaminationSystem].[dbo].[Exam\_Questions] eq

INNER JOIN [ITIExaminationSystem].[dbo].[Question\_Bank] qb

ON eq.Question\_ID = qb.Question\_ID

inner join exam e

on eq.Exam\_ID = e.Exam\_ID

inner join

Course c

on e.Course\_ID = c.Course\_ID

WHERE eq.Exam\_ID = @ExamNumber

ORDER BY qb.Question\_ID;

-- Second result set: Choices only

SELECT

qb.Question\_ID,

qc.Question\_Choice\_ID,

qc.Choice\_Text,

ROW\_NUMBER() OVER (PARTITION BY qb.Question\_ID ORDER BY qc.Question\_Choice\_ID) as ChoiceNumber

FROM [ITIExaminationSystem].[dbo].[Exam\_Questions] eq

INNER JOIN [ITIExaminationSystem].[dbo].[Question\_Bank] qb

ON eq.Question\_ID = qb.Question\_ID

LEFT JOIN [ITIExaminationSystem].[dbo].[Question\_Choice] qc

ON qb.Question\_ID = qc.Question\_ID

WHERE eq.Exam\_ID = @ExamNumber

ORDER BY

qb.Question\_ID,

qc.Question\_Choice\_ID;

END

* 1. Report 6:

**GetStudentExamAnswers**

Create PROCEDURE [dbo].[GetStudentExamAnswers]

@Exam\_ID INT,

@Student\_ID INT

AS

BEGIN

SELECT

q.Question\_ID,

q.Question\_Description,

q.Question\_Type,

q.Question\_Model\_Answer,

sea.Student\_Answer,

sea.Student\_Grade,

sea.Exam\_ID,

sea.Student\_ID,

concat(s.student\_fname, ' ', s.Student\_Lname) as StudentName,

c.course\_name

FROM [ITIExaminationSystem].[dbo].[Question\_Bank] q

INNER JOIN [ITIExaminationSystem].[dbo].[Student\_Exam\_Answer] sea

ON q.Question\_ID = sea.Question\_ID

inner join

Student s

on s.student\_ID = sea.Student\_ID

inner join Exam e

on e.Exam\_ID = sea.Exam\_ID

inner join course c

on c.course\_ID = e.Course\_ID

WHERE sea.Exam\_ID = @Exam\_ID

AND sea.Student\_ID = @Student\_ID

ORDER BY q.Question\_ID;

END