

CONTINUING EDUCATION PROGRAM

CENTER FORCOMPUTING AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERINGUNIVERSITY OF INDONESIA

SCHOOL INFORMATION SYSTEM

Member :

1. Muhammad Hudya Ramadhana
2. Mutia Ayu Dianita
3. Radityo Harimurti

Class : 1SC2 (PNJ)

Group : 2

Faculty : M. Octaviano Pratama S.St

**PROJECT ON**

School Information System

Developed by

1. Muhammad Hudya Ramadhana
2. Mutia Ayu Dianita
3. Radityo Harimurti

NIIT

**School Information System**

Batch Code : (1SC2)

Start Date : December 3rd, 2015

End Date : December 22nd, 2015

Name of the Coordinator: M. Octaviano Pratama S.ST

Name of Developer :

1. Muhammad Hudya Ramadhana
2. Mutia Ayu Dianita
3. Radityo Harimurti

Date of Submission : December 23rd,2015

**NIIT**

**CERTIFICATE**

This is to certify that this report titled School Information System embodies the original work done by Muhammad Hudya Ramadhana, Mutia Ayu Dianita, and Radityo Harimurti in partial fulfillment of their course requirement at NIIT.

Coordinator :

M. Octaviano Pratama S.ST

Writer have benefited a lot from the feedback and suggestions given to us by **Mr. M. Octaviano Pratama S.St** and other faculty members. Writer make this project to fulfill our assignment.

Writer make this project with see the basic of School Information System and make it with our own style.

Writer said Alhamdulillah to Allah S.W.T because of his blessing, writer can complete this project.

Writer know this paper is far from perfection. Writer hope if writer friend and writer faculty give any comment and suggestion to make this paper better than before.

Writer also said thankyou to :

1. Our Family who give many support to make this paper completed.
2. Mr. M. Octaviano Pratama S.St as the guide’s lecturer that give put together with the suggestion, idea & constructive spirit to make this project better.
3. Our Friends who give many idea, spirit and comment to our project.

If the writer made a mistake in writing the application of this School Information System, Writer ask forgive to all reader and listener of this paper.

Depok, December 20th 2015

**ACKNOWLEDGEMENT**

**System Summary:** SIS or School Information System is a system to keep the record of the student enrolling in this school.

**System Process:** SIS will keep information such as the student as in their id, name, sex, address and phone number. The information can be accessed by admin and the student with a password.

The student entity is connected to the class they follow which consist of id at the class, name of the class and id of the headclass. Each Class has majoring consist of id\_majoring and name of major. Each Major has course consist of id course , and name which will be input by admin. Each student will pay fee to the Treasurer. Class will be teach by teacher with id of the subject teacher, treasurer, admin will be generailized in the employee which consist of name, birhtdate, sex, id\_employee, address, phone number.

**SYSTEM ANALYSIS**

**Database Name:** *SIS*

**Number of Schemas:** 4

**Schema Names:**

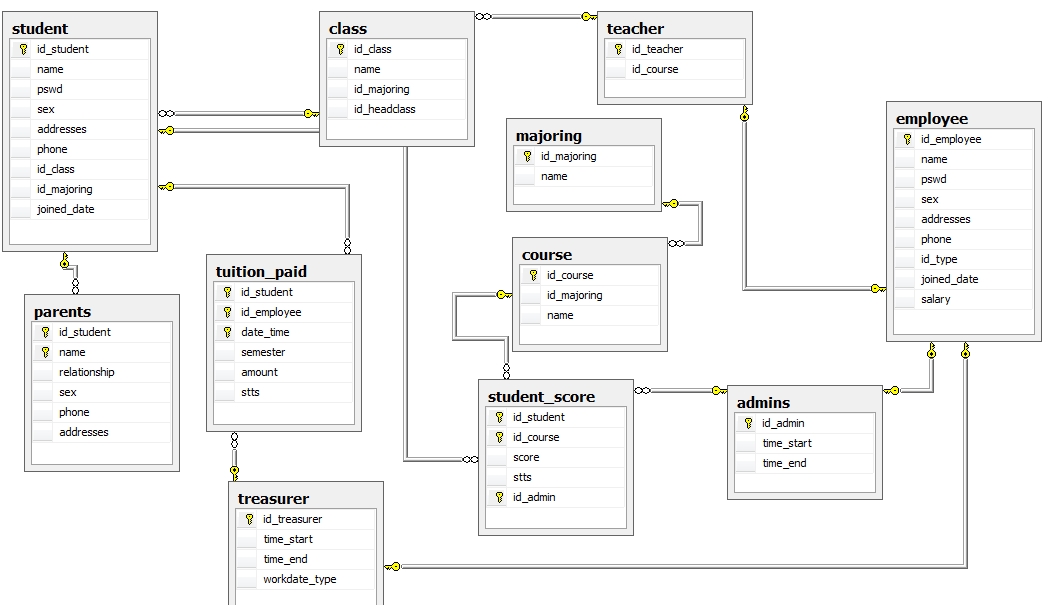
1. *StudentRelationship*
2. *Staff*
3. *TuitionPayment*
4. *StudentInformation*

**Number of tables:** 11

**Table Names:**

1. *StudentRelationship.student*
2. *StudentRelationship.parents*
3. *Staff.employee*
4. *Staff.teacher*
5. *Staff.admin*
6. *Staff.treasurer*
7. *TuitionPayment.tuition\_paid*
8. *StudentInformation.student\_score*
9. *StudentInformation.class*
10. *Student\_information.course*
11. *Student\_information.majoring*

**DATABASE DESIGN**

**

**SCHEMATIC DIAGRAM OF THE DATABASE**

**Database Name: SIS**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_student* | *char* | *5* | *Identification number of student* |
| *Name* | *varchar* | *50* | *Student name* |
| *Pswd* | *varchar* | *50* | *Student password* |
| *Sex* | *char* | *1* | *Student sex* |
| *Addresses* | *text* |  | *Student address* |
| *Phone* | *varchar* | *12* | *Student phone number* |
| *id\_class* | *char* | *5* | *Identification number of class* |
| *id\_majoring* | *char* | *5* | *Identification number of majoring* |
| *joined\_date* | *date* |  | *Student joined date* |

*StudentRelationship.student Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_student* | *char* | *5* | *Identification number of student* |
| *name* | *varchar* | *50* | *Parent name* |
| *relationship* | *varchar* | *50* | *Relationship between parent and student* |
| *sex* | *char* | *1* | *Parent sex* |
| *phone* | *varchar* | *50* | *Parent phone number* |
| *addresses* | *text* |  | *Parent address* |

*StudentRelationship.parents Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_student* | *char* | *5* | *Identification number of student* |
| *id\_employee* | *char* | *5* | *Identification number of Employee* |
| *date\_time* | *datetime* |  | *Payment date* |
| *semester* | *char* | *2* | *Student semester* |
| *amount* | *int* |  | *Amount of payment* |
| *stts* | *char* | *1* | *Payment status* |

*TuitionPayment.tuition\_paid Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_class* | *char* | *5* | *Identification number of Class* |
| *name* | *varchar* | *50* | *Class name* |
| *id\_majoring* | *char* | *5* | *Identification number of Majoring* |
| *id\_headclass* | *char* | *5* | *Identification number of Head class* |

*StudentInformation.class Table*

**TABLE DESIGN**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_majoring* | *Char* | *5* | *Identification number of Majoring* |
| *name* | *Varchar* | *50* | *Majoring name* |

*StudentInformation.majoring Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_course* | *char* | *5* | *Identification number of course* |
| *id\_majoring* | *char* | *5* | *Identification number of Majoring* |
| *name* | *varchar* | *50* | *Course name* |

*StudentInformation.course Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_admin* | *char* | *5* | *Identification number of admin* |
| *time\_start* | *time* |  | *Time of Admin start working* |
| *time\_end* | *time* |  | *Time of Admin end working* |

*Staff.admins Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_treasurer* | *char* | *5* | *Identification number of treasurer* |
| *time\_start* | *time* |  | *Time of Admin start working* |
| *time\_end* | *time* |  | *Time of Admin end working* |
| *workdate\_type* | *char* | *1* | *Workdate type od treasurer* |

*Staff.treasurer Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_student* | *char* | *5* | *Identification number of student* |
| *id\_course* | *char* | *5* | *Identification number of course* |
| *score* | *int* |  | *Student score of a course* |
| *stts* | *char* | *1* | *Status of student* |
| *id\_admin* | *char* | *5* | *Identification number of admin* |

*StudentInformation.student\_score Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_employee ,* | *char* | *5* | *Identification number of teacher* |
| *name* | *varchar* | *50* | *Employee name* |
| *pswd* | *varchar* | *50* | *Employee password* |
| *sex* | *char* | *1* | *Employee sex* |
| *addresses* | *text* |  | *Employee address* |
| *phone* | *varchar* | *12* | *Employee phone number* |
| *id\_type* | *char* | *1* | *Work type of treasurer* |
| *joined\_date* | *date* |  | *Joined date of Employee* |
| *salary* | *int* |  | *Employee salary* |

*Staff.employee Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_teacher* | *char* | *5* | *Identification number of teacher* |
| *id\_course* | *char* | *5* | *Identification number of course* |

*teacher Table*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Field Name*** | ***Data Type*** | ***Width*** | ***Description*** |
| *id\_teacher* | *char* | *5* | *Identification number of teacher* |
| *id\_course* | *char* | *5* | *Identification number of course* |

*Staff.teacher Table*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_student must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_student should be entered by S followed by 4 characters* | *Identity specification* |
| *sex should be entered by ‘M’ or ‘F’* | *CHECK constraint* |
| *id\_class is a foreign key from class table* | *Foreign key clause in the ALTER TABLE* |
| *id\_majoring is a foreign key from majoring table* | *Foreign key clause in the ALTER TABLE* |
| *Phone should be entered in the format REPLICATE('[0-9]',0-12)* | *CHECK constraint* |

*StudentRelationship.student*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_student and name should be not be left blank* | *NOT NULL clause in the CREATE TABLE statement* |
| *id\_student is a foreign key from student table* | *Foreign key clause in the ALTER TABLE* |
| *sex should be entered by ‘M’ or ‘F’* | *CHECK constraint* |
| *phone should be entered in the format*  *REPLICATE('[0-9]',0-12)* | *CHECK constraint* |

*StudentRelationship.parents*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_student,id\_employee, date\_time should be not be left blank* | *NOT NULL clause in the CREATE TABLE statement* |
| *id\_student is a foreign key from student table* | *Foreign key clause in the ALTER TABLE* |
| *id\_employee is a foreign key from student table* | *Foreign key clause in the ALTER TABLE* |
| *stts should be entered by ‘Y’ or ‘N’* | *CHECK constraint* |
| *If amount filled by 250000, ensure that the stts filled by ‘Y’* | *Trigger* |

*TuitionPayment.tuition\_paid*

**VALIDATIONS PERFORMED**

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_class must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_class should be filled by C followed by 4 characters* | *CHECK constraint* |
| *id\_majoring is a foreign key from majoring table* | *Foreign key clause in the ALTER TABLE* |
| *id\_headclass is a foreign key from student table* | *Foreign key clause in the ALTER TABLE* |

*StudentInformation.class*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_majoring must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_majoring should be filled by M followed by 4 characters* | *CHECK constraint* |

*StudentInformation.majoring*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_course must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_course should be filled by P followed by 4 characters* | *CHECK constraint* |
| *id\_majoring is a foreign key from majoring table* | *Foreign key clause in the ALTER TABLE* |

*StudentInformation.course*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_admins must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_admins should be filled by E followed by 4 characters* | *CHECK constraint* |
| *id\_admins is a foreign key from employee table* | *Foreign key clause in the ALTER TABLE* |

*Staff.admin*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_treasurer must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_treasurer should be filled by E followed by 4 characters* | *CHECK constraint* |
| *id\_treasurer is a foreign key from employee table* | *Foreign key clause in the ALTER TABLE* |

*Staff.treasurer*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_student, id\_course, and id\_admin should be not be left blank* | *NOT NULL clause in the CREATE TABLE statement* |
| *id\_student is a foreign key from student table* | *Foreign key clause in the ALTER TABLE* |
| *id\_course is a foreign key from course table* | *Foreign key clause in the ALTER TABLE* |
| *id\_admin is a foreign key from admins table* | *Foreign key clause in the ALTER TABLE* |
| *If score>=75, ensure that the stts filled by ‘Y’* | *Trigger* |
| *If score<75, ensure that the stts filled by ‘N’* | *Trigger* |

*StudentInformation.student\_score*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_employee must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_employee should be filled by E followed by 4 characters* | *CHECK constraint* |
| *id\_type should be filled by ‘1’ or ‘2’ or ‘3’* | *CHECK constraint* |
| *sex should be filled by ‘M’ or ‘F’* | *CHECK constraint* |
| *phone should be entered in the format*  *REPLICATE('[0-9]',0-12)* | *CHECK constraint* |
| *If id\_type=1, salary filled by 5000000* | *Trigger* |
| *If id\_type=2, salary filled by 4500000* | *Trigger* |
| *If id\_type=3, salary filled by 3500000* | *Trigger* |

*Staff.employee*

|  |  |
| --- | --- |
| ***Validation Required*** | ***Method(s) Used For Validation*** |
| *id\_teacher must be unique* | *Primary key clause in the CREATE TABLE statement* |
| *id\_teacher should be filled by E followed by 4 characters* | *CHECK constraint* |
| *id\_teacher is a foreign key from employee table* | *Foreign key clause in the ALTER TABLE* |
| *id\_course is a foreign key from course table* |  |

*Staff.teacher*

-- CREATE DATABASE --

create database SIS;

-- USE DATABASE --

use SIS;

-- CREATE TABLE --

create table student(

id\_student char(5) not null primary key,

name varchar (50),

pswd varchar (50),

sex char(1),

addresses text,

phone varchar (12),

id\_class char (5),

id\_majoring char (5),

joined\_date date,

);

create table parents (

id\_student char(5) not null,

name varchar(50) not null,

relationship varchar(50),

sex char (1),

phone varchar (12),

addresses text

primary key (id\_student,name)

);

create table tuition\_paid (

id\_student char(5) not null,

id\_employee char(5) not null,

date\_time datetime not null,

semester char (2),

amount int,

stts char (1),

primary key (id\_student,id\_employee,date\_time)

);

create table class(

id\_class char(5)primary key,

name varchar (50),

id\_majoring char (5),

id\_headclass char(5)

);

create table majoring(

id\_majoring char(5) primary key,

name varchar (50),

);

create table course(

id\_course char(5) primary key,

id\_majoring char (5),

name varchar(50)

);

create table admins (

id\_admin char(5) primary key,

time\_start time,

time\_end time

);

create table treasurer (

id\_treasurer char(5) primary key,

time\_start time,

time\_end time,

workdate\_type char (1)

);

create table student\_score (

id\_student char (5) not null,

id\_course char (5) not null,

score int,

stts char (1),

id\_admin char (5) not null

primary key (id\_student,id\_course,id\_admin)

);

create table teacher(

id\_teacher char(5) not null primary key,

id\_course char(5)

);

create table employee (

id\_employee char (5) primary key,

name varchar (50),

pswd varchar (50),

sex char (1),

addresses text,

phone varchar (13),

id\_type char (1),

joined\_date date,

salary int

);

**SCRIPT**

create table course(

id\_course char(5) primary key,

id\_majoring char (5),

name varchar(50)

);

create table admins (

id\_admin char(5) primary key,

time\_start time,

time\_end time

);

create table treasurer (

id\_treasurer char(5) primary key,

time\_start time,

time\_end time,

workdate\_type char (1)

);

create table student\_score (

id\_student char (5) not null,

id\_course char (5) not null,

score int,

stts char (1),

id\_admin char (5) not null

primary key (id\_student,id\_course,id\_admin)

);

create table teacher(

id\_teacher char(5) not null primary key,

id\_course char(5)

);

create table employee (

id\_employee char (5) primary key,

name varchar (50),

pswd varchar (50),

sex char (1),

addresses text,

phone varchar (13),

id\_type char (1),

joined\_date date,

salary int

);

create table course(

id\_course char(5) primary key,

id\_majoring char (5),

name varchar(50)

);

create table admins (

id\_admin char(5) primary key,

time\_start time,

time\_end time

);

create table treasurer (

id\_treasurer char(5) primary key,

time\_start time,

time\_end time,

workdate\_type char (1)

);

create table student\_score (

id\_student char (5) not null,

id\_course char (5) not null,

score int,

stts char (1),

id\_admin char (5) not null

primary key (id\_student,id\_course,id\_admin)

);

create table teacher(

id\_teacher char(5) not null primary key,

id\_course char(5)

);

create table employee (

id\_employee char (5) primary key,

name varchar (50),

pswd varchar (50),

sex char (1),

addresses text,

phone varchar (13),

id\_type char (1),

joined\_date date,

salary int

);

--CONSTRAINT--

--constraint student

alter table student drop constraint chk\_pswd

alter table student add constraint chk\_id\_student check (id\_student like 'S%%%%');

alter table student add constraint chk\_sex check (sex like 'M' OR sex like 'F');

alter table student add constraint chk\_phone\_student check (phone like REPLICATE('[0-9]',0-12))

--constraint class

alter table class add constraint chk\_class check(id\_class like 'C%%%%');

--constraint majoring

alter table majoring add constraint chk\_major check(id\_majoring like 'M%%%%');

--constraint employee

alter table employee add constraint id\_employee check (id\_employee like 'E%%%%');

alter table employee add constraint id\_type check (id\_type like '1' OR id\_type like '2' OR id\_type like '3');

alter table employee add constraint chk\_sex\_emp check (sex like 'M' OR sex like 'F');

alter table employee add constraint chk\_phone\_employee check(phone like REPLICATE('[0-9]',0-12))

--constraint course

alter table course drop constraint chk\_course

alter table course add constraint chk\_course check(id\_course like 'P%%%%');

--constraint admins

alter table admins add constraint chk\_Ad check(id\_admin like 'E%%%%');

--coinstraint parents

alter table parents add constraint chk\_id\_student2 check (id\_student like 'S%%%%');

alter table parents add constraint chk\_sex\_Ad check (sex like 'M' OR sex like 'F');

alter table parents add constraint chk\_phone\_parent check(phone like REPLICATE('[0-9]',0-12))

--constraint tuition paid

alter table tuition\_paid add constraint chk\_status check(stts like 'Y' OR stts like 'N');

--- FK ----------

alter table parents add foreign key (id\_student) references student (id\_student)

alter table course add foreign key (id\_majoring) references majoring (id\_majoring)

alter table student add foreign key (id\_class) references class (id\_class)

alter table class add foreign key (id\_majoring) references majoring (id\_majoring)

alter table class add foreign key (id\_headclass) references teacher (id\_teacher)

alter table teacher add foreign key (id\_teacher) references employee (id\_employee)

alter table Admins add foreign key (id\_admin) references employee (id\_employee)

alter table treasurer add foreign key (id\_treasurer) references employee (id\_employee)

alter table tuition\_paid add foreign key (id\_student) references student (id\_student)

alter table tuition\_paid add foreign key (id\_employee) references treasurer (id\_treasurer)

alter table student\_score add foreign key (id\_course) references course (id\_course)

alter table student\_score add foreign key (id\_student) references student (id\_student)

alter table student\_score add foreign key (id\_admin) references Admins (id\_admin)

--SELECT--

select \* from student;

select \* from student\_score;

select \* from course;

select \* from parents;

select \* from teacher;

select \* from class;

select \* from majoring;

select \* from teacher;

select \* from tuition\_paid;

select \* from employee;

select \* from admins;

---Trigger--

--Check Score

create trigger tr\_chk\_score on dbo.student\_score

after insert,update

as

declare @score int

begin

select @score = score from inserted

if (@score > 75)

update student\_score set stts = 'Y'

else

if (@score < 75)

update student\_score set stts = 'N'

end

--Check Paid

create trigger tr\_chk\_paid on dbo.tuition\_paid

after insert,update

as

declare @amount int

begin

select @amount = amount from inserted

if (@amount = 250000 )

update tuition\_paid set stts = 'Y'

else

update tuition\_paid set stts = 'N'

end

--Chk Salary

create trigger tr\_chk\_salary on dbo.employee

after insert,update

as

declare @id\_type char (1)

begin

select @id\_type = id\_type from inserted

if (@id\_type = 1)

update employee set salary = 5000000

else

if (@id\_type = 2)

update employee set salary = 4500000

else

if(@id\_type = 3)

update employee set salary = 3500000

end

-------PROCEDURE-----------

---procedure auto generate id student--

drop procedure autospecidstudent

create procedure autospecidstudent

@name varchar (50),

@pswd varchar (50),

@sex char(1),

@addresses text,

@phone varchar (12),

@id\_class char (5),

@id\_majoring char (5),

@joined\_date date

as

declare @id\_student char(5)

if exists(select id\_student from student where id\_student = 'S0001')

begin

select @id\_student = MAX(right(id\_student ,3)) from student

select @id\_student =

case

when @id\_student >=0 and @id\_student <10 then 'S000' + CONVERT(char(5),@id\_student + 1)

when @id\_student >=10 and @id\_student <100 then 'S00' + CONVERT(char(5),@id\_student + 1)

when @id\_student >=100 and @id\_student <1000 then 'S0' + CONVERT(char(5),@id\_student + 1)

end

end

else

update student set id\_student = 'S0001'

insert into student values(@id\_student,@name,@pswd,@sex,@addresses,@phone,@id\_class,@id\_majoring,@joined\_date)

go

exec autospecidstudent 'Octav','greget123','M','Bogor','083239123','C0001','M0001','12-12-2015'

exec autospecidstudent 'Waluyo','greget123','M','Bogor','083239123','C0001','M0001','12-12-2015'

-------auto generate id employee

select \* from student

drop procedure autospecidemployee

create procedure autospecidemployee

@name varchar (50),

@pswd varchar (50),

@sex char(1),

@addresses text,

@phone varchar (12),

@id\_type char (1),

@salary int,

@joined\_date date

as

declare @id\_employee char(5)

if exists(select id\_employee from employee where id\_employee = 'E0001')

begin

select @id\_employee = MAX(right(id\_employee ,3)) from employee

select @id\_employee =

case

when @id\_employee >=0 and @id\_employee <10 then 'E000' + CONVERT(char(5),@id\_employee + 1)

when @id\_employee >=10 and @id\_employee <100 then 'E00' + CONVERT(char(5),@id\_employee + 1)

when @id\_employee >=100 and @id\_employee <1000 then 'E0' + CONVERT(char(5),@id\_employee + 1)

end

end

else

update employee set id\_employee = 'E0001'

insert into employee values(@id\_employee,@name,@pswd,@sex,@addresses,@phone,@id\_type,@joined\_date,@salary)

go

exec autospecidemployee 'Aban','fchr123','M','Bogor','0891823','2','','12-12-2015'

select \* from employee

-------auto generate id class

select \* from class

drop procedure autospecidclass

create procedure autospecidclass

@name varchar (50),

@id\_majoring char (5),

@id\_heacdlass char (5)

as

declare @id\_class char(5)

if exists(select id\_class from class where id\_class = 'C0001')

begin

select @id\_class = MAX(right(id\_class ,3)) from class

select @id\_class =

case

when @id\_class >=0 and @id\_class <10 then 'C000' + CONVERT(char(5),@id\_class + 1)

when @id\_class >=10 and @id\_class <100 then 'C00' + CONVERT(char(5),@id\_class + 1)

when @id\_class >=100 and @id\_class <1000 then 'C0' + CONVERT(char(5),@id\_class + 1)

end

end

else

insert into class values('C0001',@name,@id\_majoring,@id\_heacdlass)

insert into class values(@id\_class,@name,@id\_majoring,@id\_heacdlass)

go

exec autospecidclass 'X IPA 2','M0001','E0001'

select \* from class

---auto generate id majoring

create procedure autospecidmajoring

@name varchar (50)

as

declare @id\_majoring char(5)

if exists(select id\_majoring from majoring where id\_majoring = 'M0001')

begin

select @id\_majoring = MAX(right(id\_majoring ,3)) from majoring

select @id\_majoring =

case

when @id\_majoring >=0 and @id\_majoring <10 then 'M000' + CONVERT(char(5),@id\_majoring + 1)

when @id\_majoring >=10 and @id\_majoring <100 then 'M00' + CONVERT(char(5),@id\_majoring + 1)

when @id\_majoring >=100 and @id\_majoring <1000 then 'M0' + CONVERT(char(5),@id\_majoring + 1)

end

end

else

insert into majoring values('M0001',@name)

insert into majoring values(@id\_majoring,@name)

go

exec autospecidmajoring 'Arsitektur'

select \* from course

----- auto spec id course

create procedure autospecidcourse

@id\_majoring char (5),

@name varchar (50)

as

declare @id\_course char(5)

if exists(select id\_course from course where id\_course = 'P0001')

begin

select @id\_course = MAX(right(id\_course ,3)) from course

select @id\_course =

case

when @id\_course >=0 and @id\_course <10 then 'P000' + CONVERT(char(5),@id\_course + 1)

when @id\_course >=10 and @id\_course <100 then 'P00' + CONVERT(char(5),@id\_course + 1)

when @id\_course >=100 and @id\_course <1000 then 'P0' + CONVERT(char(5),@id\_course + 1)

end

end

else

insert into course values('P0001',@id\_majoring,@name)

insert into course values(@id\_course,@id\_majoring,@name)

go

exec autospecidcourse 'M0001','Bahasa Jerman'

----- VIEW ------

create view vscore as

select A.id\_student as id\_s,A.name, B.score as score, B.stts as statuss,

C.name as name\_course from student A, student\_score B, course C where

A.id\_student = B.id\_student AND C.id\_course = B.id\_course

select id\_s,name,name\_course,score,statuss from vscore where statuss = 'N'

select id\_s,name,name\_course,score,statuss from vscore where statuss = 'Y'

select id\_s,name,name\_course,score,statuss from vscore where score > 78

create view vpayment as

select A.id\_student as id\_s, A.name,B.semester ,B.amount,B.stts as status

from student A, tuition\_paid B where A.id\_student = B.id\_student

select id\_s,name,semester,amount,status from vpayment where semester = 1

select id\_s,name,semester,amount,status from vpayment where status = 'Y'

select id\_s,name,semester,amount,status from vpayment where status = 'N'

create view vstudent\_status as

select A.id\_student as id\_s, A.name,B.name as Class, C.name as Majoring

from student A, class B, majoring C

select name,class,majoring from vstudent\_status where Majoring = 'IPA'

use SIS

GRANT CREATE PROCEDURE,CREATE TABLE,CREATE FUNCTION

to mutiad

use SIS

GRANT DELETE,INSERT,SELECT,UPDATE

to raditt

create user raditt

for login radit

create user mutiad

for login mutia

---BACKUP

BACKUP DATABASE SIS

to disk= 'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DHANA\MSSQL\Backup\SIS.bak'

WITH FORMAT

GO

BACKUP LOG SIS

to disk= 'C:\Program Files\Microsoft SQL Server\MSSQL10\_50.DHANA\MSSQL\Backup\SIS.bak'

GO

**REPORTS OUTLINE**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Report Name*** | ***Report Type*** | ***Description*** | ***Tables/Queries Used*** |
| vscore | Query Output | This report displays the student score details such as the id student, name of student, score, status, and name of course for a given status and score | student, student\_course and course |
| vpayment | Query Output | This report displays the student’s tuition payment details such as the name, semester ,amount, and status for a given semester and status | student, tuition\_paid |
| vstudent\_status | Query Output | This report displays the student’s status /information details such as name, class, and majoring for a given majoring name | student, class, majoring |

**CONFIGURATION**

**Hardware** : *IntelR CoreTM i3-3217U CPU @ 1.80 GHz (4CPUs), 2.00GB RAM, 500 GB HDD, ASUS X450C 802.11b/g/n,6 cell Li-ion battery*

**Operating System** : *Windows 7 Ultimate*

**Software** : *Microsoft SQL Server 2008 R2 Express*

**PROJECT FILE DETAILS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **File Name** | **Remarks** |
| 1 | *SIS.mdf*  *SIS\_log.ldf* | *SQL Server databasenthat contains tables, procedures, triggers, constraints, and queries* |