



Views_Test

vw_QuestionPool

Purpose:

Combines questions with their course IDs so you can see which questions belong to which course.

Test_Query

```
Select* from vw_QuestionPool
```

RESULT

Results Messages

	Quest_ID	Quest_type	Quest_text	degree	Crs_id
1	43	Multiple Choice	Which keyword is used to filter rows in SQL?	5	29
2	44	Multiple Choice	Which constraint enforces uniqueness?	5	29
3	45	True & False	A primary key can contain NULL.	5	29
4	46	True & False	FOREIGN KEY ensures referential integrity.	5	30
5	47	Text	Define normalization in DB design (short).	10	30
6	48	Text	Explain what an index does.	10	29



Views_Test

vw_StudentResults

Purpose:

Shows student results per course including pass/fail status.

Test_Query

```
Select * From vw_StudentResults
```

RESULT

Results		Messages				
	St_ID	St_Fname	St_Lname	Crs_name	final_degree	is_passed
1	61	Student	01	SQL Fundamentals	15	1
2	62	Student	02	SQL Fundamentals	10	1
3	61	Student	01	Database Design	10	1
4	66	Student	06	Python for Data	5	0



Views_Test

vw_TextAnswersReview

Purpose:

Displays only text-type question answers for review.

Test_Query

```
Select * From vw_TextAnswersReview
```

RESULT

Results Messages

	St_ID	Quest_Text	Ans_Text	is_correct
1	61	Define normalization in DB design (short).	Where	1



Views_Test

vw_Courses

Purpose:

Simple view listing course information

Test_Query

```
Select * From vw_Courses
```

RESULT

<div><div><div><div></div></div></div><div>Results</div></div> <div><div><div></div></div></div> <div>Messages</div>
--



Views_Test

vw_CourseInstructorByYear

Purpose:

Shows which instructor taught which course in which intake and year.

Test_Query

```
Select * From vw_CourseInstructorByYear
```

RESULT

 Results  Messages

	Crs_name	Fname	Ins_year	intake_name
1	SQL Fundamentals	Ali	2026	Intake-45-DE-Cairo
2	Database Design	Ali	2026	Intake-45-DE-Cairo
3	Python for Data	Mona	2026	Intake-45-AI-Alex
4	ETL Concepts	Mona	2026	Intake-45-AI-Alex



Views_Test

vw_StudentsDistribution

Purpose:

Counts students grouped by branch + track + intake.

Test_Query

```
Select * From vw_StudentsDistribution
```

RESULT

Results		Messages		
	branch_name	track_name	intake_name	TotalStudents
1	Alex	AI	Intake-45-AI-Alex	5
2	Cairo	DE	Intake-45-DE-Cairo	5



Views_Test

vw_Intakes

Purpose:

Shows intake information with branch and track context.

Test_Query

```
Select * From vw_Intakes
```

RESULT

Results		Messages				
	Intk_ID	Intake_Name	start_year	end_year	branch_name	track_name
1	14	Intake-45-DE-Cairo	2025-10-01	2026-06-30	Cairo	DE
2	15	Intake-45-AI-Alex	2025-10-01	2026-06-30	Alex	AI



Views_Test

vw_ExamDetails

Purpose:

Provides detailed info about exams with organizational context.

Test_Query

```
Select * From vw_ExamDetails
```

RESULT

Results Messages							
Exam_id	Exam_type	branch_name	track_name	intake_name	Start_time	End_time	Total_time
13	Final	Cairo	DE	Intake-45-DE-Cairo	10:00:00.0000000	12:00:00.0000000	120
14	Quiz	Cairo	DE	Intake-45-DE-Cairo	13:00:00.0000000	14:00:00.0000000	60
15	Final	Alex	AI	Intake-45-AI-Alex	11:00:00.0000000	13:00:00.0000000	120



Views_Test

vw_ExamQuestions

Purpose:

Provides detailed info about exams with organizational context.

Test_Query

```
select * from vw_ExamQuestions
```

RESULT

Results		Messages	
	Exam_id	Quest_text	degree
1	13	Which keyword is used to filter rows in SQL?	5
2	13	Which constraint enforces uniqueness?	5
3	13	A primary key can contain NULL.	5
4	13	Explain what an index does.	10
5	14	FOREIGN KEY ensures referential integrity.	5
6	14	Define normalization in DB design (short).	10
7	15	Which keyword is used to filter rows in SQL?	5
8	15	FOREIGN KEY ensures referential integrity.	5



Views_Test

10 vw_InstructorCourses

Purpose:

Provides detailed info about exams with organizational context.

Test_Query

```
Select * from vw_InstructorCourses
```

RESULT



Results



Messages

	Fname	Crs_name
1	Ali	SQL Fundamentals
2	Ali	Database Design
3	Mona	Python for Data
4	Mona	ETL Concepts



Constraints Test

CK_User_Role

Purpose:

Ensure user role is only:

'Student'

'Instructor'

'Training_Manager'

Test_Query

```
INSERT INTO [User] ( User_Name, Role,Is_active,Pass)
VALUES ( 'wrong_user', 'Viewer',1,1);
```

RESULT

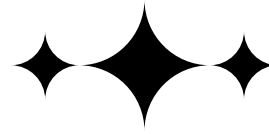
Messages

Msg 547, Level 16, State 0, Line 2
The INSERT statement conflicted with the CHECK constraint "CK_User_Role". The conflict occurred in database "ProjectDE_EXAM", table "dbo.User", column 'Role'.
The statement has been terminated.

Completion time: 2026-02-20T16:45:25.3447829+02:00



Constraints Test



UQ_User_User_Name

Purpose:

Prevent duplicate usernames.

Test_Query

```
INSERT INTO [User] ( User_Name, Role,Is_active,Pass)
VALUES ( 'amir_kholosy', 'Student',1,1);
INSERT INTO [User] ( User_Name, Role,Is_active,Pass)
VALUES ( 'amir_kholosy', 'Student',0,1);
```

RESULT

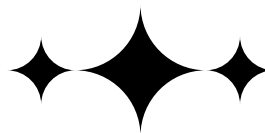
Messages

Msg 2627, Level 14, State 1, Line 3
Violation of UNIQUE KEY constraint 'UQ_User_Name'. Cannot insert duplicate key in object 'dbo.User'. The duplicate key value is (amir_kholosy).
The statement has been terminated.

Completion time: 2026-02-20T16:50:34.5465558+02:00



Constraints Test



CK_Course_Degree (Min <= Max)

Purpose:

Ensure minimum degree does not exceed maximum degree.

////////////////////////////////////

Test_Query

```
INSERT INTO Course ( Crs_Name, Min_Degree, Max_Degree)
VALUES ( 'Python', 120, 100);
```

////////////////////////////////////

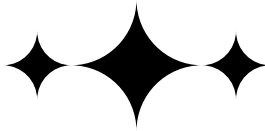
RESULT

```
Msg 547, Level 16, State 0, Line 1
The INSERT statement conflicted with the CHECK constraint "CK_Course_Degree". The conflict occurred in database "ProjectDE_EXAM", table "dbo.Course".
The statement has been terminated.
```

```
Completion time: 2026-02-20T16:52:52.6780814+02:00
```



Constraints Test



FK_Student_User

Purpose:

Ensure every student must have a valid User account.

Test_Query

```
INSERT INTO Student ( Usr_ID, Email,Track_ID,St_Fname,St_Lname,Mngr_ID,Intk_ID,Brn_ID)
VALUES ( 9999, 'student2@test.com',20,'asad','8aba',27,14,15);
```

RESULT

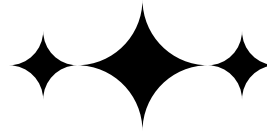
Messages

Msg 547, Level 16, State 0, Line 1
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Student_User". The conflict occurred in database "ProjectDE_EXAM", table "dbo.User", column 'Usr_ID'.
The statement has been terminated.

Completion time: 2026-02-20T16:59:49.4483172+02:00



Constraints Test



CK_Exam_Dates

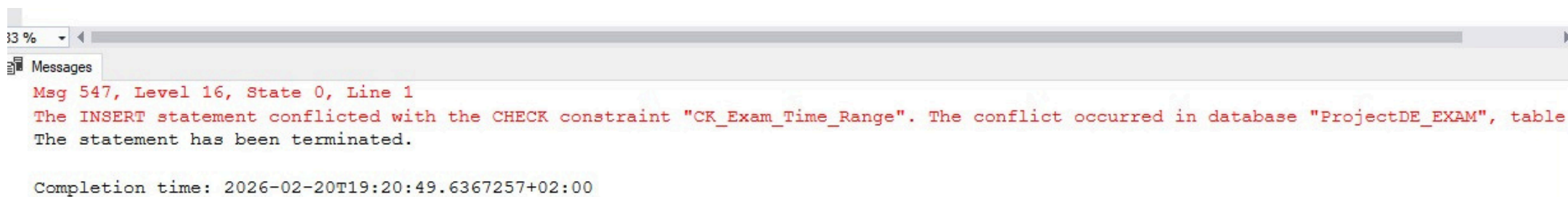
Purpose:

Ensure exam end time is after start time.

Test_Query

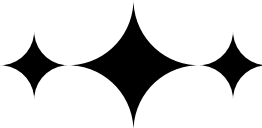
```
INSERT INTO Exam ( Exam_Type, Start_Time, End_Time ,Ins_ID,Total_Time,Exam_Date,Year,Allowance_Option,Intk_ID,Crs_ID)
VALUES ( 'Final', '12:00:00.000000', '10:00:00.000000',20,60,'2026-02-20',2020,1,14,29);
```

RESULT





Constraints Test



CK_Stud_Ans_Is_Correct

Purpose:

Ensure Is_Correct only accepts 0 or 1.

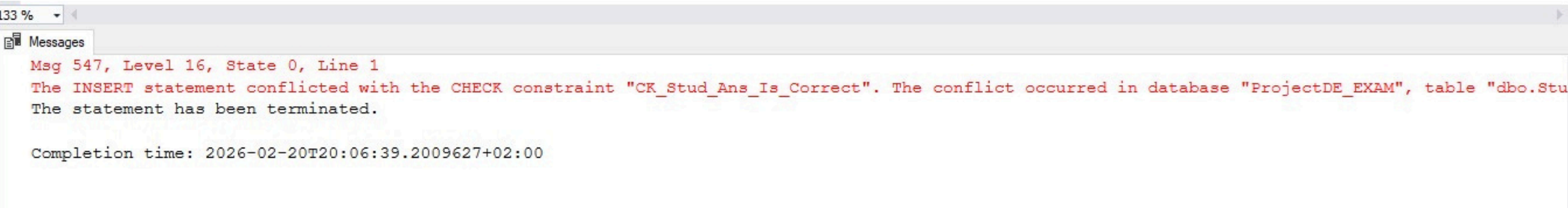
////////////////////////////////////

Test_Query

```
INSERT INTO Stud_Ans ( Is_Correct,Exam_ID,Ans_Text,Degree_given,Quest_ID,St_ID)
VALUES (5,13,'Where',15,43,61);
```

////////////////////////////////////

RESULT





SP_Tests

SubmitAnswer

Purpose:

Saves student answer, checks correctness depending on question type (MCQ / TF / Text), calculates degree, and inserts record. Prevents submission if exam time is closed.



RESULT

```
SELECT dbo.fn_IsExamAvailable(13) AS Exam13;
SELECT dbo.fn_IsExamAvailable(14) AS Exam14;
SELECT dbo.fn_IsExamAvailable(15) AS Exam15;
UPDATE Exam
SET Exam_Date = CAST(GETDATE() AS DATE),
    Start_Time = '00:00',
    End_Time = '23:59'
WHERE Exam_id = 13;
SELECT dbo.fn_IsExamAvailable(13);

EXEC SubmitAnswer
    @Stud_id = 62,
    @Exam_id = 13,
    @Question_id = 48,
    @Answer_text = 'Index speeds up data retrieval by providing a fast lookup structure.';
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-21T15:21:17.9499162+02:00



SP_Tests

CalculateResult

Purpose:

Calculates total student score in exam and inserts final result into Course_Result.

RESULT

```
EXEC CalculateResult  
@Stud_id = 62,  
@Exam_id = 13,  
@Crs_id = 29;
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-21T15:22:59.0949859+02:00



SP_Tests

AddCourse

Purpose:

Saves student answer, checks correctness depending on question type (MCQ / TF / Text), calculates degree, and inserts record. Prevents submission if exam time is closed.

RESULT

```
test_procedure.sql...G5K\eLROWAD (68))* X procedure_proj2.s...G5K\eLROWAD (57)) SQLQuery3.sql - D...G5K\eLROWA
-- 1 Test AddCourse success
EXEC AddCourse
    @Crs_name = 'Machine Learnings',
    @Crs_desc = 'Intro to ML3',
    @Max_degree = 100,
    @Min_degree = 50;

-- 2 Test AddCourse duplicate
EXEC AddCourse
    @Crs_name = 'Machine Learning',
    @Crs_desc = 'Intro to ML',
    @Max_degree = 100,
    @Min_degree = 50;

-- 3 Test AddCourse invalid degree
EXEC AddCourse
    @Crs_name = 'Deep Learning',
    @Crs_desc = 'DL Basics',
    @Max_degree = 50,
    @Min_degree = 60; -- should fail
```

118 %

Messages

Msg 50000, Level 16, State 1, Procedure AddCourse, Line 15 [Batch Start Line 1]
Course already exists

Msg 50000, Level 16, State 1, Procedure AddCourse, Line 22 [Batch Start Line 1]
Min degree must be less than Max degree

Completion time: 2026-02-20T15:14:14.2106124+02:00



SP_Tests

AddInstructor

Purpose:

Adds instructor if email is unique.

RESULT

```
DESKTOP-UDKBG5K\..._EXAM - dbo.User    DESKTOP-UDKBG5K\M...- dbo.Instructor    test_prosedure.sq...G5K\eLROWAD (6)

EXEC AddInstructor
    @F_name = 'Khalid',
    @L_name = 'Mohamed',
    @email = 'khalid.ins@ex.com',
    @user_id = 158; -- ins_ali already exists, let's pick a new user_id in your

-- 2 Duplicate email test
EXEC AddInstructor
    @F_name = 'Ahmad',
    @L_name = 'Ali',
    @email = 'khalid.ins@ex.com',
    @user_id = 142;

-- 3 User role wrong
EXEC AddInstructor
    @F_name = 'Sara',
    @L_name = 'Hassan',
    @email = 'sara.ins@ex.com',
    @user_id = 138; -- tm_ahmed is Training_Manager

-- 4 User already linked
EXEC AddInstructor
    @F_name = 'Ali',
    @L_name = 'Mostafa',
    @email = 'ali.ins2@ex.com',
    @user_id = 141; -- already linked to instructor

-----
-- 1 Success

118 %

Messages
Msg 50000, Level 16, State 1, Procedure AddInstructor, Line 16 [Batch Start Line 26]
Instructor email already exists
Msg 50000, Level 16, State 1, Procedure AddInstructor, Line 34 [Batch Start Line 26]
User is not assigned as Instructor
Msg 50000, Level 16, State 1, Procedure AddInstructor, Line 41 [Batch Start Line 26]
User already linked to another instructor

Completion time: 2026-02-20T15:33:15.5339042+02:00
```



SP_Tests

AddStudent

Purpose:

Adds student record with references to intake, branch, track, manager.

RESULT

```
-- 1 Success
EXEC AddStudent
    @F_name = 'Ahmed',
    @L_name = 'Khaled',
    @Email = 'ahmed.st@ex.com',
    @user_id = 155, -- st_06
    @mngr_id = 27,
    @IntakeID = 14,
    @branchID = 14,
    @track_id = 10;

-- 2 Duplicate email
EXEC AddStudent
    @F_name = 'Ali',
    @L_name = 'Mostafa',
    @Email = 'ahmed.st@ex.com',
    @user_id = 150,
    @mngr_id = 27,
    @IntakeID = 14,
    @branchID = 14,
    @track_id = 10;

-- 3 Wrong role
EXEC AddStudent
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-20T16:55:19.6741371+02:00



SP_Tests

AddStudent

Purpose:

Adds student record with references to intake, branch, track, manager.

RESULT

```
-- 2 Duplicate email
EXEC AddStudent
    @F_name = 'Ali',
    @L_name = 'Mostafa',
    @Email = 'ahmed.st@ex.com',
    @user_id = 150,
    @mngr_id = 27,
    @IntakeID = 14,
    @branchID = 14,
    @track_id = 10;

-- 3 Wrong role
EXEC AddStudent
    @F_name = 'Sara',
    @L_name = 'Hassan',
    @Email = 'sara.st@ex.com',
    @user_id = 138, -- tm_ahmed is Training_Manager
    @mngr_id = 27,
    @IntakeID = 14,
    @branchID = 14,
    @track_id = 10;

-- 4 Invalid branch
EXEC AddStudent
    @F_name = 'Mona',
    @L_name = 'Kamal',
    @Email = 'mona.st@ex.com',
    @user_id = 151,
    @mngr_id = 28,
```

118 %

Messages

Msg 50000, Level 16, State 1, Procedure AddStudent, Line 20 [Batch Start Line 64]
Student email already exists

Msg 50000, Level 16, State 1, Procedure AddStudent, Line 38 [Batch Start Line 64]
User is not assigned as Student

Msg 50000, Level 16, State 1, Procedure AddStudent, Line 45 [Batch Start Line 64]
User already linked to another student

Completion time: 2026-02-20T16:56:24.7445363+02:00



SP_Tests

AddIntake

Purpose:

Creates intake batch with date validation.

RESULT

```
-- 1 Success
EXEC AddIntake
    @intake_name = 'Intake-46-DE-Cairo',
    @start_year  = '2026-10-01',
    @end_year    = '2027-06-30',
    @training_mgr = 27,
    @branchID    = 14,
    @track_id    = 10;

-- 2 Invalid years
EXEC AddIntake
    @intake_name = 'Intake-47-DE-Cairo',
    @start_year  = '2027-06-30',
    @end_year    = '2027-01-01',
    @training_mgr = 27,
    @branchID    = 14,
    @track_id    = 10;

-- 3 Duplicate intake name
EXEC AddIntake
    @intake_name = 'Intake-45-DE-Cairo',
    @start_year  = '2025-10-01',
    @end_year    = '2026-06-30',
    @training_mgr = 27,
    @branchID    = 14,
    @track_id    = 10;
```

118 %

Messages

Msg 50000, Level 16, State 1, Procedure AddIntake, Line 18 [Batch Start Line 99]
Invalid intake years

Msg 50000, Level 16, State 1, Procedure AddIntake, Line 46 [Batch Start Line 99]
Intake already exists for this branch

Completion time: 2026-02-20T16:57:07.0389663+02:00



SP_Tests

AddBranch

Purpose:

Adds branch only if same name doesn't exist in same department.

RESULT

```
-- 1 Success
EXEC AddBranch
    @branch_name = 'Giza',
    @dept_id = 16,
    @mngr_id = 27;

-- 2 Duplicate
EXEC AddBranch
    @branch_name = 'Cairo',
    @dept_id = 16,
    @mngr_id = 27;

-- 3 Invalid dept
EXEC AddBranch
    @branch_name = 'Sharm',
    @dept_id = 999,
    @mngr_id = 27;

-- 1 Success
EXEC AddTrack
    @track_name = 'Cyber Security',
    @mngr_id = 28;
```

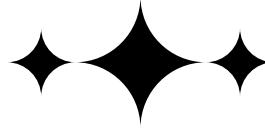
118 %
Messages

```
Msg 50000, Level 16, State 1, Procedure AddBranch, Line 28 [Batch Start Line 126]
Branch already exists in this department
Msg 50000, Level 16, State 1, Procedure AddBranch, Line 14 [Batch Start Line 126]
Department does not exist
```

Completion time: 2026-02-20T16:57:55.6930821+02:00



SP_Tests



AddTrack

Purpose:

Creates track if name not already used.



RESULT



SP_Tests

CreateExamRandom

Purpose:

Creates exam + randomly selects questions by type.

RESULT

```
-- Random Exam Test
```

```
EXEC CreateExamRandom
```

```
  @Exam_type = 'Final',
```

```
  @Exam_date = '2026-02-25',
```

```
  @Start_time = '10:00',
```

```
  @End_time   = '12:00',
```

```
  @Total_time = 120,
```

```
  @Allowance_option = 1,
```

```
  @Year = 2026,
```

```
  @Crs_id = 29,
```

```
  @inst_id = 20,
```

```
  @intakeID = 14,
```

```
  @Num_MCQ = 2,
```

```
  @Num_TF = 1,
```

```
  @Num_Text = 1;
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-20T17:15:45.4232082+02:00



SP_Tests

10 CreateExamManual

Purpose:

Creates exam if not exists, then adds one question manually.

////////////////////////////////////

RESULT

```
-- Manual Exam Test
EXEC CreateExamManual
    @Exam_type = 'Quiz',
    @Exam_date = '2026-02-28',
    @Start_time = '13:00',
    @End_time   = '14:00',
    @Total_time = 60,
    @Allowance_option = 0,
    @Year = 2026,
    @Crs_id = 30,
    @inst_id = 21,
    @intakeID = 15,
    @Question_id = 46;

-- Test creating new users
EXEC sp_CreateUser @Username = 'tm_test', @Password = '12345', @
EXEC sp_CreateUser @Username = 'st_test', @Password = '12345', @
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-20T17:16:24.1754567+02:00



Purpose:

.....

```
-- Test creating new users
EXEC sp_CreateUser @Username = 'tm_test', @Password = '12345', @Role = 'Training_Manager',@isactive = True;
EXEC sp_CreateUser @Username = 'st_test', @Password = '12345', @Role = 'Student',@isactive = True;
EXEC sp_CreateUser @Username = 'ins_test', @Password = '12345', @Role = 'Instructor',@isactive = True;
-----

-- Successful login
EXEC sp_Login @Username = 'tm_test', @Password = '12345';

-- Invalid login
```

118 %

Messages

Commands completed successfully.

Completion time: 2026-02-20T15:21:13.7676883+02:00



SP_Tests

sp_Login

Purpose:

Checks login credentials.

RESULT

```
-- Successful login
EXEC sp_Login @Username = 'tm_test', @Password = '12345';

-- Invalid login
EXEC sp_Login @Username = 'tm_test', @Password = 'wrongpass';
EXEC sp_Login @Username = 'nonexistent', @Password = '12345';
```

118 %
Results Messages

	Result
1	Login Successful

	Result
1	Invalid Credentials

	Result
1	Invalid Credentials