1. Insert new student and his score in exam in different subjects as transaction.

begin;

INSERT INTO `iti`.`student` VALUES ('7', 'ola', 'hatem', 'ola@gmail.com', 'female', '2000-05-01', '1');

INSERT INTO `iti`.`exam` VALUES ('7', '1', '2023-03-25', '100');

INSERT INTO `iti`.`exam` VALUES ('7', '2', '2023-03-25', '90');

commit;  
2. Display the date of exam as the following: day 'month name' year.

select concat(day(date), ' ', monthname(date), ' ', year(date)) as date from exam;  
3. Display name and age of each students

select fname, (year(CURDATE()) - year(birthdate)) as age from student;  
4. Display the name of students with their Rounded score in each Exam

select fname, round(score) as roundedScore

from student, exam where id = student\_id;  
5. Display the name of students with the year of Birthdate

*select fname, year(birthdate) as year from student;*6. Add new exam result, in date column use NOW

INSERT INTO exam VALUES ('5', '2', now(), '80.5');  
7. Create Hello world function which take username and return welcome message to user using his name.

DELIMITER //

CREATE FUNCTION iti.hello\_world(username VARCHAR(50))

RETURNS VARCHAR(100)

BEGIN

DECLARE message VARCHAR(100);

SET message = CONCAT('Welcome, ', username);

RETURN message;

END//

DELIMITER ;

Select iti.hello\_world("yasmine");  
8. Create multiply function which take two number and return the multiply of them

DELIMITER //

CREATE FUNCTION iti.multiply(a INT, b INT)

RETURNS INT

BEGIN

DECLARE result INT;

SET result = a \* b;

RETURN result;

END//

DELIMITER ;

Select iti.multiply(2, 3);  
9. Create function which takes student id and Exam id and return score the student in Exam.

DELIMITER $$

CREATE FUNCTION iti.get\_ score (student\_id INT, subject\_id INT)

RETURNS INT

BEGIN

DECLARE result INT;

SELECT score INTO result FROM iti.exam

WHERE student\_id = student\_id AND subject\_id = subject\_id;

RETURN result;

END$$

DELIMITER ;

select iti.get\_ score (4, 2);  
10. Create function which takes Exam id and return the number of students who failed in an Exam (Score less than 50).

DELIMITER $$

CREATE FUNCTION failed\_students(exam\_id INT)

RETURNS INT

BEGIN

DECLARE result INT;

SELECT COUNT(\*) INTO result FROM exam

WHERE subject\_id = exam\_id AND score < 50;

RETURN result;

END$$

DELIMITER ;

select failed\_students(1);  
11. Create function which take subject name and return the average of max grades for subject

DELIMITER $$

CREATE FUNCTION avg\_max\_grades(subject\_name VARCHAR(255))

RETURNS FLOAT

BEGIN

DECLARE result FLOAT;

SELECT AVG(max\_score) INTO result FROM (

SELECT MAX(score) AS max\_score FROM exam

JOIN subject ON exam.subject\_id = subjects.id

WHERE subjects.name = subject\_name

GROUP BY exam.student\_id

) AS max\_grades;

RETURN result;

END$$

DELIMITER ;  
12. Create Table called Deleted\_Students which will hold the deleted students info (same columns as in student tables)

CREATE TABLE `iti`.`deleted\_student` (

`id` INT NOT NULL,

`fname` VARCHAR(45) NULL,

`lname` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`gender` ENUM('fname', 'lname') NULL,

`birthdate` DATE NULL,

`track\_id` INT NULL,

PRIMARY KEY (`id`),

INDEX `track\_id\_fk\_idx` (`track\_id` ASC) VISIBLE,

CONSTRAINT `track\_id\_fk`

FOREIGN KEY (`track\_id`)

REFERENCES `iti`.`track` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);  
13. Create trigger to save the deleted student from Student table to  
Deleted\_Students.

DELIMITER $$

CREATE TRIGGER save\_deleted\_student

AFTER DELETE ON student

FOR EACH ROW

BEGIN

INSERT INTO deleted\_student (id, fname, lname, email, gender, birthdate, track\_id)

VALUES (OLD.id, OLD.fname, OLD.lname, OLD.email, OLD.gender, OLD.birthdate, OLD.track\_id);

END$$

DELIMITER ;  
14. Create trigger to save the newly added students to Student table to  
Backup\_Students.

CREATE TABLE `iti`.`backup\_student` (

`id` INT NOT NULL,

`fname` VARCHAR(45) NULL,

`lname` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`gender` ENUM('fname', 'lname') NULL,

`birthdate` DATE NULL,

`track\_id` INT NULL,

PRIMARY KEY (`id`),

INDEX `trackID\_fk\_idx` (`track\_id` ASC) VISIBLE,

CONSTRAINT `trackID\_fk`

FOREIGN KEY (`track\_id`)

REFERENCES `iti`.`track` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

DELIMITER $$

CREATE TRIGGER save\_new\_student

AFTER INSERT ON student

FOR EACH ROW

BEGIN

INSERT INTO backup\_student (id, fname, lname, email, gender, birthdate, track\_id)

VALUES (NEW.id, NEW.fname, NEW.lname, NEW.email, NEW.gender, NEW.birthdate, NEW.track\_id);

END$$

DELIMITER ;  
15. (Bouns) Create trigger to keep track the changes of contact info table  
(add/update rows); it will logs the time of action and description of  
action to another table.  
16. Dump your database (Grading Database) into SQL file.

mysql -u root -p

mysqldump -u root iti > student.sql  
17. Dump Students table into file.

mysqldump -u root iti student > student\_dump\_file.sql  
18. Import SQL file into your backup database (Grading\_Backup Database)

mysql -u root ITI students2 < students\_dump\_file.sql