DAY3 MYSQL

1-BEGIN

INSERT INTO Student (firstName, lastName, birthdate, Address, Email, gender, trackId, subjectId)

VALUES (Ahmed, 'Ali', '2000-01-01', 'Cairo', 'ahmed22@gmail.com', 'male', 1, 1);

SET newStudentId = LAST\_INSERT\_ID();

INSERT INTO studentSubject (studentId, subjectId, examScore) VALUES

(newStudentId, 1, 85),

(newStudentId, 2, 90);

COMMIT;

2- SELECT studentSubject

CONCAT(DAY(examDate), '-', MONTHNAME(examDate), '-', YEAR(examDate)) FROM studentSubject;

3-

SELECT firstName, lastName,

FLOOR(DATEDIFF(CURDATE(), birthdate) / 365) AS age

FROM student;

4- SELECT

CONCAT(s.firstName,’ ’,s.lastName)AS Full Name,ROUND(ss.examScore) AS Rounded examScore

FROM Student s

JOIN studentSubject ss

ON s.studentId=ss.studentId;

5- SELECT CONCAT(firstName,’ ’,lastName)AS Full Name, YEAR(birthdate) AS yearOfBirth FROM Student ;

6-

INSERT INTO studentSubject (studentId, subjectId, examScore, examId,examDate)

VALUES (1, 2, 85,1, NOW());

-7

DELIMITER //

CREATE FUNCTION helloUser (username VARCHAR(50))

RETURNS VARCHAR (100)

RETURN CONCAT ('Welcome ', username) AS Welcome Message //

-8 DELIMITER //

CREATE FUNCTION multiplyNumbers )num1 INT, num2 INT)

RETURNS INT

RETURN (num1 \* num2) //

9- DELIMITER //

CREATE FUNCTION examscore (exam\_id INT , student\_id INT)

RETURNS INT

BEGIN

DECLARE score INT;

SELECT examScore INTO score from studentSubject

where studentId =student\_id AND examId =exam\_id;

RETURN score;

END //

10- DELIMITER //

CREATE FUNCTION countStudentsFailedInExam (exam\_id INT)

RETURNS INT

BEGIN

DECLARE counter INT;

SELECT COUNT(examScore) INTO counter FROM studentSubject WHERE examId = exam\_id AND examScore < 50;

RETURN counter;

END //

11- DELIMITER //

CREATE FUNCTION averageMaxGrade(subject\_name VARCHAR(100))

RETURNS DECIMAL(5, 2)

BEGIN

DECLARE avgMaxGrade DECIMAL(5, 2);

SELECT AVG(maxScore) INTO avgMaxGrade

FROM subjects

WHERE Name = subject\_name;

RETURN avgMaxGrade;

END //

DELIMITER //

-12

CREATE TABLE deletedStudents (

studentId INT PRIMARY KEY AUTO\_INCREMENT,

firstName VARCHAR(50),

lastName VARCHAR(50),

birthdate DATE,

Address VARCHAR(255),

Email VARCHAR(100),

gender ENUM('male', 'female'),

trackId INT,

subjectId INT,

deleted\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (trackId) REFERENCES Tracks(trackId),

FOREIGN KEY (subjectId) REFERENCES Subjects(subjectId)

);

13-

DELIMITER //

CREATE TRIGGER afterStudentDelete

AFTER DELETE ON StudeCnt

FOR EACH ROW

BEGIN

INSERT INTO deletedStudents (

studentId, firstName, lastName,

birthdate,Address, Email,

gender, trackId, subjectId, deletedAt) VALUES (

OLD.studentId, OLD.firstName, OLD.lastName,

OLD.birthdate, OLD.Address, OLD.Email,

OLD.gender, OLD.trackId, OLD.subjectId,

NOW());

END //

14-CREATE TABLE backupStudents (

studentId INT PRIMARY KEY,

firstName VARCHAR(50),

lastName VARCHAR(50),

birthdate DATE,

Address VARCHAR(255),

Email VRCHAR(100),

gender ENUM('male', 'female'),

trackId INT,

subjectId INT,

backupAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

DELIMITER //

CREATE TRIGGER afterStudentInsert

AFTER INSERT ON students

FOR EACH ROW

BEGIN

INSERT INTO backupStudents (

studentId,

studentId, firstName, lastName,

birthdate,Address, Email,

gender, trackId, subjectId, backupAt)

VALUES (

NEW.studentId, NEW.firstName, NEW.lastName,

NEW.birthdate, NEW.Address, NEW.Email,

NEW.gender, NEW.trackId, NEW.subjectId,

NOW( );

END //

16- mysqldump -u ubuntu -p iti> ITI.sql

17- mysqldump -u root -p iti Student > studentsTableDump.sql

18-

CREATE DATABASE gradingBackup;

mysql -u root -p gradingBackup < studentsTableDump.sql

REMAINING FROM DAY2

6- BEGIN

INSERT INTO Student (firstName, lastName, birthdate, Address, Email, gender, trackId, subjectId)

VALUES (Ahmed, 'Ali', '2000-01-01', 'Cairo', 'ahmed22@gmail.com', 'male', 1, 1);

SET newStudentId = LAST\_INSERT\_ID();

INSERT INTO studentSubject (studentId, subjectId, examScore) VALUES

(newStudentId, 1, 85),

(newStudentId, 2, 90);

COMMIT;

17- CREATE TABLE Track (

trackName VARCHAR (50),

trackId INT PRIMARY KEY AUTO\_INCREMENT );

* INSERT INTO Track (trackName) VALUES
* (‘frontend’),
* (‘backend’),
* (‘full stack php’),
* (‘full stack .net’),
* (‘social media’);

-

ALTER TABLE Student

ADD COLUMN trackId INT,

ADD CONSTRAINT fk\_Track FOREIGN KEY (trackId) REFERENCES Track (trackId) ON DELETE CASCADE;

* UPDATE Student SET trackId =5 WHERE studentId =1;
* UPDATE Student SET trackId =4 WHERE studentId =2;
* …………. To 5 row

-CREATE VIEW studentTrackView AS

SELECT s.first\_name, s.last\_name, t.track\_name

FROM Student s

JOIN Track t ON s.track\_id = t.track\_id;

SELECT \* FROM studentTrackView;

18- ALTER TABLE Subject

ADD COLUMN trackId INT,

ADD CONSTRAINT fk\_Track FOREIGN KEY (trackId) REFERENCES Track (trackId) ON DELETE CASCADE;

* UPDATE Subject SET trackId =5 WHERE subjectId =1;
* UPDATE Subject SET trackId =4 WHERE subjectId =2;
* ………….5 row

-CREATE VIEW subjectTrackView AS

SELECT s.name, t.track\_name

FROM Subject s

JOIN tracks t ON s.track\_id = t.track\_id;

SELECT \* FROM subjectTrackView;

19--CREATE VIEW studentSubjectView AS

SELECT stu.firstName, stu.lastName, sub.name

FROM Student stu

JOIN studentSubject ss ON stu.studentId = ss.studentId

JOIN Subject sub ON ss.subjectId=sub.subjectId;

SELECT \* FROM studentSubjectView;

20---CREATE VIEW studentSubjectScoreView AS

SELECT stu.firstName, stu.lastName, sub.name

FROM Student stu

JOIN studentSubject ss ON stu.studentId = ss.studentId

JOIN Subject sub ON ss.subjectId=sub.subjectId;

SELECT \* FROM studentSubjectScoreView

21-

CREATE TEMPORARY TABLE tempSubject AS

SELECT name as subjectName, maxScore

FROM Subject;

SELECT \* FROM tempSubject;

22- DELETE FROM Student

WHERE studentId IN (

SELECT studentId

FROM studentSubject

WHERE subjectId = 1 AND examScore < 50

);