**SQL**

**Create and insert Assignment**

**Exercise 2.1:**

Ans:

create table Trainer\_Info

(Trainer\_Idvarchar(20),

Salutation varchar(7),

Trainer\_Namevarchar(30),

Trainer\_Locationvarchar(30),

Trainer\_Trackvarchar(15),

Trainer\_Qualificationvarchar(100),

Trainer\_Experianceint(11),

Trainer\_Emailvarchar(100),

Trainer\_Passwordvarchar(20)

);

Ans:

create table Batch\_Info

(Batch\_Idvarchar(20),

Batch\_Ownervarchar(30),

Batch\_BU\_Namevarchar(30)

);

Ans:

create table Module\_Info(

Module\_Idvarchar(20),

Module\_Namevarchar(40),

Module\_Durationinteger(11)

);

Ans:

create table Associate\_Info(

Associate\_Idvarchar(20),

Salutation varchar(7),

Associate\_Namevarchar(30),

Associate\_Locationvarchar(30),

Associate\_Trackvarchar(15),

Associate\_Qualificationvarchar(200),

Associate\_Emailvarchar(100),

Associate\_Passowordvarchar(20)

);

Ans:

create table Questions(

Question\_Idvarchar(20),

Module\_Idvarchar(20),

Question\_Textvarchar(900)

);

Ans:

create table Associate\_status(

Associate\_Idvarchar(20),

Module\_Idvarchar(20),

Start\_Datevarchar(20),

End\_Datevarchar(20),

AFeedbackGivenvarchar(20),

TFeedbackGivenvarchar(20)

);

Ans:

create table Trainer\_Feedback(

Trainer\_Idvarchar(20),

Question\_Idvarchar(20),

Batch\_Idvarchar(20),

Module\_Idvarchar(20),

Trainer\_RatingInt(11)

);

Ans:

create table Associate\_Feedback(

Associate\_Idvarchar(20),

Question\_Idvarchar(20),

Module\_Idvarchar(20),

Associate\_RatingInt(11)

);

Ans:

create table Login\_Details(

User\_Idvarchar(20),

User\_Passwordvarchar(20)

);

**Exercise 2.2:**

Problem Statement:

Modify the table Associate\_Status to include following two columns:

Column Name: Batch\_Id

Details: Batch Id from batch\_Info

Data Types: Character

Size: 20

Column Name: Trainer\_Id

Details: Trainer Id from trainer\_Info

Data Types: Character

Size: 20

Modify two columns data types of the table Associate\_Status as follows:

Start\_Date

Desired Type: Date

End\_Date

Desired Type: Date

Ans:

Alter table Associate\_Status

Add Batch\_Idvarchar(20),

add Trainer\_Idvarchar(20),

modify Start\_Date date,

modify End\_Date date;

**Exercise 2.3**

Problem Statement:

Insertbelow details into table:

Trainer\_Info

Trainer\_Id,Salutation,Trainer\_Name,Trainer\_Location,Trainer\_Track,Trainer\_Qualification,Trainer\_Experiance,Trainer\_Email,Trainer\_Password

F001,Mr.,JOHN,Toronto,Java,Bachelor of Technology,12,John@alliance.com,fac1@123

F002,Mr.,Mike ,Chicago,DotNet,Bachelor of Technology,12,Mike@alliance.com,fac2@123

F003,Mr.,Jacob,Washington,Mainframe,Bachelor of Technology,10,Jacob@alliance.com,fac3@123

F004,Mrs.,Teena,Mexico,Java,Master of Computer Applications,9,Teena@alliance.com,fac4@123

F005,Miss.,Jeni,Boston,Testing,Master of Computer Applications,6,Jeni@alliance.com,fac5@123

F006,Mr., William, New York City,Mainframe,Bachelor of Technology,9, William @alliance.com,fac6@123

F007,Ms.,Jessi,Los Angeles,Testing,Bachelor of Technology,5,Jessi@alliance.com,fac7@123

F008,Mr.,Ryan , New York City,Java,Master of Science In Information Technology,12,Ryan@alliance.com,fac8@123

Ans:

insert into Trainer\_Infovalues('F001','Mr.','JOHN','Toronto','Java','Bachelor of Technology',12,'John@alliance.com','fac1@123'),

('F002','Mr.','Mike' ,'Chicago','DotNet','Bachelor of Technology',12,'Mike@alliance.com','fac2@123'),

('F003','Mr.','Jacob','Washington','Mainframe','Bachelor of Technology',10,'Jacob@alliance.com','fac3@123'),

('F004','Mrs.','Teena','Mexico','Java','Master of Computer Applications',9,'Teena@alliance.com','fac4@123'),

('F005','Miss.','Jeni','Boston','Testing','Master of Computer Applications',6,'Jeni@alliance.com','fac5@123'),

('F006','Mr.', 'William', 'New York City','Mainframe','Bachelor of Technology',9, 'William @alliance.com','fac6@123'),

('F007','Ms.','Jessi','Los Angeles','Testing','Bachelor of Technology',5,'Jessi@alliance.com','fac7@123'),

('F008','Mr.','Ryan' , 'New York City','Java','Master of Science In Information Technology',12,'Ryan@alliance.com','fac8@123');

Batch\_Info

Batch\_Id,Batch\_Owner,Batch\_BU\_Name

B001,MRS.Bella,MSP

B002,MRS.Alexa,HEALTHCARE

B003,MR.Luke,LIFE SCIENCES

B004,MR.Carter,BFS

B005,MR.Jack,COMMUNICATIONS

B006,MRS.Lucy,RETAIL & HOSPITALITY

B007,MRS.Eva,MSP

B008,MRS.Maria,BPO

B009,MR.Henry,MSP

Ans:

insert into Batch\_Info values('B001','MRS.Bella','MSP'),

('B002','MRS.Alexa','HEALTHCARE'),

('B003','MR.Luke','LIFE SCIENCES'),

('B004','MR.Carter','BFS'),

('B005','MR.Jack','COMMUNICATIONS'),

('B006','MRS.Lucy','RETAIL & HOSPITALITY'),

('B007','MRS.Eva','MSP'),

('B008','MRS.Maria','BPO'),

('B009','MR.Henry','MSP');

Module\_Info

Module\_Id,Module\_Name,Module\_Duration

O10SQL,Oracle 10g SQL ,16

O10PLSQL,Oracle 10g PL/ SQL ,16

J2SE,Core Java SE 1.6,288

J2EE,Advanced Java EE 1.6,80

JAVAFX,JavaFX 2.1,80

DOTNT4,.Net Framework 4.0 ,50

SQL2008,MSSQl Server 2008,120

MSBI08,MS BI Studio 2008,158

SHRPNT,MS Share Point ,80

ANDRD4,Android 4.0,200

EM001,Instructor,0

EM002,Course Material,0

EM003,Learning Effectiveness,0

EM004,Environment,0

EM005,Job Impact,0

TM001,Attendees,0

TM002,Course Material,0

TM003,Environment,0

Ans:

insert into Module\_Infovalues('O10SQL','Oracle 10g SQL',16),

('O10PLSQL','Oracle 10g PL/ SQL' ,16),

('J2SE','Core Java SE 1.6',288),

('J2EE','Advanced Java EE 1.6',80),

('JAVAFX','JavaFX 2.1',80),

('DOTNT4','.Net Framework 4.0' ,50),

('SQL2008','MS SQl Server 2008',120),

('MSBI08','MS BI Studio 2008',158),

('SHRPNT','MS Share Point' ,80),

('ANDRD4','Android 4.0',200),

('EM001','Instructor',0),

('EM002','Course Material',0),

('EM003','Learning Effectiveness',0),

('EM004','Environment',0),

('EM005','Job Impact',0),

('TM001','Attendees',0),

('TM002','Course Material',0),

('TM003','Environment',0);

Associate\_Info

Associate\_Id,Salutation,Associate\_Name,Associate\_Location,Associate\_Track,Associate\_Qualification,Associate\_Email,Associate\_Password

A001,Miss.,Julia,Toronto,Java,Bachelor of Technology,Julia@hp.com,tne1@123

A002,Mrs.,Anna,Chicago,Java,Bachelor of Engineering In Information Technology,Anna@cognizant.com,tne2@123

A003,Mr.,Daniel,Washington,Java,Bachelor of Engineering In Computers,daniel@ibm.com,tne3@123

A004,Mr.,Andrew,New York City, DotNet, Master of Computer Applications,Andrew@finolex.com,tne4@123

A005,Miss.,Sofia,Chicago,Mainframe,Bachelor of Engineering In Information Technology,Sofia@microsoft.com,tne5@123

A006,Mrs.,Lily,Toronto,Testing,Master of Computer Applications,Lily@cognizant.com,tne6@123

A007,Mr.,Mason, New York City,Java,Bachelor of Technology,Mason@honda.com,tne7@123

Ans:

insert into Associate\_Infovalues('A001','Miss.','Julia','Toronto','Java','Bachelor of Technology','Julia@hp.com','tne1@123'),

('A002','Mrs.','Anna','Chicago','Java','Bachelor of Engineering In Information Technology','Anna@cognizant.com','tne2@123'),

('A003','Mr.','Daniel','Washington','Java','Bachelor of Engineering In Computers','daniel@ibm.com','tne3@123'),

('A004','Mr.','Andrew','New York City','DotNet','Master of Computer Applications','Andrew@finolex.com','tne4@123'),

('A005','Miss.','Sofia','Chicago','Mainframe','Bachelor of Engineering In Information Technology','Sofia@microsoft.com','tne5@123'),

('A006','Mrs.','Lily','Toronto','Testing','Master of Computer Applications','Lily@cognizant.com','tne6@123'),

('A007','Mr.','Mason', 'New York City','Java','Bachelor of Technology','Mason@honda.com','tne7@123');

Questions

Question\_Id,Module\_Id,Question\_Text

Q001,EM001,Instructor knowledgeable and able to handle all your queries

Q002,EM001,All the topics in a particular course handled by the trainer without any gaps or slippages

Q003,EM002,The course materials presentation, handson, etc. referred during the training are relevant and useful.

Q004,EM002,The Hands on session adequate enough to grasp the understanding of the topic.

Q005,EM002,The reference materials suggested for each module are adequate.

Q006,EM003,Knowledge and skills presented in this training are applicable at your work

Q007,EM003,This training increases my proficiency level

Q008,EM004,The physical environment e.g. classroom space, air-conditioning was conducive to learning.

Q009,EM004,The software/hardware environment provided was sufficient for the purpose of the training.

Q010,EM005,This training will improve your job performance.

Q011,EM005,This training align with the business priorities and goals.

Q012,TM001,Participants were receptive and had attitude towards learning.

Q013,TM001,All participants gained the knowledge and the practical skills after this training.

Q014,TM002,The course materials presentation, handson, etc. available for the session covers the entire objectives of the course.

Q015,TM002,Complexity of the course is adequate for the participate level.

Q016,TM002,Case study and practical demos helpful in understanding of the topic

Q017,TM003,The physical environment e.g. classroom space, air-conditioning was conducive to learning.

Q018,TM003,The software/hardware environment provided was adequate for the purpose of the training.

Ans:

insert into Questions values('Q001','EM001','Instructor knowledgeable and able to handle all your queries'),

('Q002','EM001','All the topics in a particular course handled by the trainer without any gaps or slippages'),

('Q003','EM002','The course materials presentation, handson, etc. referred during the training are relevant and useful'),

('Q004','EM002','The Hands on session adequate enough to grasp the understanding of the topic.'),

('Q005','EM002','The reference materials suggested for each module are adequate.'),

('Q006','EM003','Knowledge and skills presented in this training are applicable at your work'),

('Q007','EM003','This training increases my proficiency level'),

('Q008','EM004','The physical environment e.g. classroom space, air-conditioning was conducive to learning.'),

('Q009','EM004','The software/hardware environment provided was sufficient for the purpose of the training.'),

('Q010','EM005','This training will improve your job performance.'),

('Q011','EM005','This training align with the business priorities and goals.'),

('Q012','TM001','Participants were receptive and had attitude towards learning.'),

('Q013','TM001','All participants gained the knowledge and the practical skills after this training.'),

('Q014','TM002','The course materials presentation, handson, etc. available for the session covers the entire objectives of the course.'),

('Q015','TM002','Complexity of the course is adequate for the participate level.'),

('Q016','TM002','Case study and practical demos helpful in understanding of the topic'),

('Q017','TM003','The physical environment e.g. classroom space, air-conditioning was conducive to learning.'),

('Q018','TM003','The software/hardware environment provided was adequate for the purpose of the training');

Associate\_Status

Associate\_Id,Module\_Id,Batch\_Id,Trainer\_Id,Start\_Date,End\_Date

A001,O10SQL,B001,F001,2000-12-15,2000-12-25

A002,O10SQL,B001,F001,2000-12-15,2000-12-25

A003,O10SQL,B001,F001,2000-12-15,2000-12-25

A001,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A002,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A003,O10PLSQL,B002,F002,2001-2-1,2001-2-12

A001,J2SE,B003,F003,2002-8-20,2002-10-25

A002,J2SE,B003,F003,2002-8-20,2002-10-25

A001,J2EE,B004,F004,2005-12-1,2005-12-25

A002,J2EE,B004,F004,2005-12-1,2005-12-25

A003,J2EE,B004,F004,2005-12-1,2005-12-25

A004,J2EE,B004,F004,2005-12-1,2005-12-25

A005,JAVAFX,B005,F006,2005-12-4,2005-12-20

A006,JAVAFX,B005,F006,2005-12-4,2005-12-20

A006,SQL2008,B006,F007,2007-6-21,2007-6-28

A007,SQL2008,B006,F007,2007-6-21,2007-6-28

A002,MSBI08,B007,F006,2009-6-26,2009-6-29

A003,MSBI08,B007,F006,2009-6-26,2009-6-29

A004,MSBI08,B007,F006,2009-6-26,2009-6-29

A002,ANDRD4,B008,F005,2010-6-5,2010-6-28

A005,ANDRD4,B008,F005,2010-6-5,2010-6-28

A003,ANDRD4,B009,F005,2011-8-1,2011-8-20

A006,ANDRD4,B009,F005,2011-8-1,2011-8-20

Ans:

insert into Associate\_Status(Associate\_Id,Module\_Id,Batch\_Id,Trainer\_Id,Start\_Date,End\_Date)

values('A002','O10SQL','B001','F001','2000-12-15','2000-12-25'),

('A003','O10SQL','B001','F001','2000-12-15','2000-12-25'),

('A001','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),

('A002','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),

('A003','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),

('A001','J2SE','B003','F003','2002-8-20','2002-10-25'),

('A002','J2SE','B003','F003','2002-8-20','2002-10-25'),

('A001','J2EE','B004','F004','2005-12-1','2005-12-25'),

('A002','J2EE','B004','F004','2005-12-1','2005-12-25'),

('A003','J2EE','B004','F004','2005-12-1','2005-12-25'),

('A004','J2EE','B004','F004','2005-12-1','2005-12-25'),

('A005','JAVAFX','B005','F006','2005-12-4','2005-12-20'),

('A006','JAVAFX','B005','F006','2005-12-4','2005-12-20'),

('A006','SQL2008','B006','F007','2007-6-21','2007-6-28'),

('A007','SQL2008','B006','F007','2007-6-21','2007-6-28'),

('A002','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A003','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A004','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A002','ANDRD4','B008','F005','2010-6-5','2010-6-28'),

('A005','ANDRD4','B008','F005','2010-6-5','2010-6-28'),

('A003','ANDRD4','B009','F005','2011-8-1','2011-8-20'),

('A006','ANDRD4','B009','F005','2011-8-1','2011-8-20');

**Exercise2.4:**

|  |
| --- |
| Hands-on Exercise Objective |
| After completing the hands-on exercises, you will be able to:  Demonstrate how to change data in table |
|  |  |

Problem Statement:

Change the password of trainer F004 from fac4@123 to nn4@123

Ans:

update Trainer\_info

SET Trainer\_Password = 'nn4@123'

WHERE Trainer\_Id = 'F004';

**Exercise2.5:**

Remove following record form associate\_status table

A003,J2EE,B004,F004,2005-12-1,2005-12-25

Ans:

delete from associate\_status

where Associate\_Id = 'A003'

and Module\_Id = 'J2EE'

and Batch\_Id = 'B004'

and Trainer\_Id = 'F004'

and Start\_Date = '2005-12-1'

and End\_Date = '2005-12-25';

**Exercise2.6**

Fetch first five trainers who have maximum years of experience & display

Ans:

select \* from trainer\_info

order by Trainer\_Experiancedesc

limit 5;

**exercise2.7:**

Problem Statement:

Begin transaction & insert below torecords intoLogin\_Details

'U001' Admin1@123

'U002' Admin2@123

Perform rollback operation & verify whether any records are inserted in table or not.

Ans:

insert into Login\_Details

values('U001','Admin1@123'),

('U002','Admin2@123');

rollback;

**Exercise2.8:**

Problem Statement:

Create a dummy user in database. Grant create & select table privilege to him/her.

Repeat the above all queries using login credentials of newly created user.

Revoke the privilege assigned to this newly created user.

Ans:

After creating the dummy\_user

Grant select, create on \*.\* to dummy\_user

Revoke select, create on \*.\* from dummy\_user

**Exercise2.9:**

Remove table Login\_Details from database.

Ans:

Drop table Login\_Details;

**Constraints Assignment**

**Exercise3.1:**

Apply all Integrity constraint as given below:

Ans:

ALTER TABLE Trainer\_Info

ADD CONSTRAINT Trainer\_Id\_pk PRIMARY KEY(Trainer\_Id),

ADD CONSTRAINT Trainer\_Id\_ckCHECK(Trainer\_Id LIKE 'F%'),

MODIFY Salutation varchar(7) NOT NULL,

MODIFY Trainer\_Namevarchar(30) NOT NULL,

MODIFY Trainer\_Locationvarchar(30) NOT NULL,

MODIFY Trainer\_Trackvarchar(15) NOT NULL,

MODIFY Trainer\_Qualificationvarchar(100) NOT NULL,

MODIFY Trainer\_Emailvarchar(100) NOT NULL,

MODIFY Trainer\_Passwordvarchar(20) NOT NULL;

Ans:

ALTER TABLE Batch\_Info

ADD CONSTRAINT Batch\_Id\_pk PRIMARY KEY(Batch\_Id),

ADD CONSTRAINT Batch\_Id\_ckCHECK(Batch\_Id LIKE 'B%'),

MODIFY Batch\_Ownervarchar(30) NOT NULL,

MODIFY Batch\_BU\_Namevarchar(30) NOT NULL;

Ans:

ALTER TABLE Module\_Info

ADD CONSTRAINT Module\_Id\_pk PRIMARY KEY(Module\_Id),

ADD CONSTRAINT Module\_Id\_ckCHECK(Module\_Id = upper(Module\_Id)),

MODIFY Module\_Namevarchar(40) NOT NULL,

MODIFY Module\_Durationinteger(11) NOT NULL;

Ans:

ALTER TABLE Associate\_Info

ADD CONSTRAINT Associate\_Id\_pk PRIMARY KEY(Associate\_Id),

ADD CONSTRAINT Associate\_Id\_ckCHECK(Associate\_Id LIKE 'A%'),

MODIFY Salutation varchar(7) NOT NULL,

MODIFY Associate\_Namevarchar(30) NOT NULL,

MODIFY Associate\_Locationvarchar(30) NOT NULL,

MODIFY Associate\_Trackvarchar(15) NOT NULL,

MODIFY Associate\_Qualificationvarchar(200) NOT NULL,

MODIFY Associate\_Emailvarchar(100) NOT NULL,

MODIFY Associate\_Passowordvarchar(20) NOT NULL;

Ans:

ALTER TABLE Questions

ADD CONSTRAINT Question\_Id\_pk PRIMARY KEY(Question\_Id),

ADD CONSTRAINT Question\_Id\_ckCHECK(Question\_Id LIKE 'Q%'),

ADD CONSTRAINT FOREIGN KEY(Module\_Id)

REFERENCES Module\_Info(Module\_Id),

MODIFY Question\_Textvarchar(900) NOT NULL;

Ans:

ALTER TABLE Associate\_Status

ADD CONSTRAINT Associate\_Id\_fk FOREIGN KEY(Associate\_Id)

REFERENCES Associate\_Info(Associate\_Id),

ADD CONSTRAINT Module\_Id\_fk FOREIGN KEY(Module\_Id)

REFERENCES Module\_Info(Module\_Id),

ADD CONSTRAINT Batch\_Id\_fk FOREIGN KEY(Batch\_Id)

REFERENCES Batch\_Info(Batch\_Id),

ADD CONSTRAINT Trainer\_Id\_fk FOREIGN KEY(Trainer\_Id)

REFERENCES Trainer\_Info(Trainer\_Id),

MODIFY Associate\_Idvarchar(20) NOT NULL,

MODIFY Module\_Idvarchar(20) NOT NULL,

MODIFY Batch\_Idvarchar(20) NOT NULL,

MODIFY Trainer\_Idvarchar(20) NOT NULL;

Ans:

ALTER TABLE Trainer\_Feedback

ADD CONSTRAINT Trainer1\_Id\_fk FOREIGN KEY(Trainer\_Id)

REFERENCES Trainer\_Info(Trainer\_Id),

ADD CONSTRAINT Question\_Id\_fk FOREIGN KEY(Question\_Id)

REFERENCES Questions(Question\_Id),

ADD CONSTRAINT Batch1\_Id\_fk FOREIGN KEY(Batch\_Id)

REFERENCES Batch\_Info(Batch\_Id),

ADD CONSTRAINT Module1\_Id\_fk FOREIGN KEY(Module\_Id)

REFERENCES Module\_Info(Module\_Id),

MODIFY Trainer\_Idvarchar(20) NOT NULL,

MODIFY Question\_Idvarchar(20) NOT NULL,

MODIFY Batch\_Idvarchar(20) NOT NULL,

MODIFY Module\_Idvarchar(20) NOT NULL,

MODIFY Trainer\_RatingInt(11) NOT NULL;

Ans:

ALTER TABLE Associate\_feedback

ADD CONSTRAINT Associate1\_Id\_fk FOREIGN KEY(Associate\_Id)

REFERENCES Associate\_Info(Associate\_Id),

ADD CONSTRAINT Question\_Id1\_fk FOREIGN KEY(Question\_Id)

REFERENCES Questions(Question\_Id),

ADD CONSTRAINT Module2\_Id\_fk FOREIGN KEY(Module\_Id)

REFERENCES Module\_Info(Module\_Id),

MODIFY Associate\_Idvarchar(20) NOT NULL,

MODIFY Question\_Idvarchar(20) NOT NULL,

MODIFY Module\_Idvarchar(20) NOT NULL,

MODIFY Associate\_RatingInt(11) NOT NULL;

**Exercise3.2:**

Consider another scenario, create following two table in database

Table

Name: product

Column: productIDint(10) primary key

Column: productnamevarchar(20)

Column: productpriceint(5) not null

Table

Name: user

Column: userIDvarchar(10) primary key

Column: productIDint(10) references product(productID)

Column: username varchar(20)

Insert below records into the product table

* productID, productname, productprice
  + 1,'A Dongle',290
  + 2,'B Dongle',1250
* productID,productname
  + 3,'C Dongle'

Now disable foreign key on user table.

Insert below records into the user table

* userID, productID, username
  + 'U001',1,'Luke'
  + 'U002',11,'Raul'

Verify whether second record is inserted properly, justify the reason if yes or not.

Enable key again on user table.

Ans:

CREATE TABLE product(

productIDint(10),

productnamevarchar(20),

productpriceint(5) NOT NULL,

CONSTRAINT productID\_pk PRIMARY KEY(productID)

);

CREATE TABLE user1(

userIDvarchar(10),

productIDint(10),

username varchar(20),

CONSTRAINT userID\_pk PRIMARY KEY(userID),

CONSTRAINT productID\_fk FOREIGN KEY(productID)

REFERENCES product(productID)

);

INSERT INTO product(productID,productname,productprice)

values(1,'A Dongle',290),

(2,'B Dongle',1250);

INSERT INTO product(productID,productname)

values(3,'C Dongle');--------[This query shows an error, because we have NOT NULL constraint]

ALTER TABLE user1

DROP CONSTRAINT productid\_fk;

INSERT INTO user1(userID,productid,username)

VALUES('U001',1,'Luke'),

('U002',11,'Raul');----------[This query will not generate any error as foreign key constraint was disabled]

ALTER TABLE user1

ADD CONSTRAINT productID\_fk FOREIGN KEY(productID)

REFERENCES product(productID);

**Operators Assignment**

**Exercise4.1:**

1. SELECT \* FROM Trainer\_Info

WHERE Trainer\_Email IS null;

2. SELECT Trainer\_Id,Trainer\_Name,Trainer\_Track,Trainer\_Location

FROM Trainer\_Info

WHERE Trainer\_Experiance> 4;

3. SELECT \* FROM Module\_Info

WHERE Module\_Duration> 200;

4.SELECT Trainer\_Id,Trainer\_Name FROM Trainer\_Info

WHERE Trainer\_Qualification NOT LIKE ‘Bachelor of Technology’;

5.SELECT \* FROM Module\_Info

WHERE Module\_Duration BETWEEN 200 AND 300;

6.SELECT Trainer\_Id,Trainer\_Name FROM Trainer\_Info

WHERE Trainer\_Name LIKE 'M%';

7. SELECT Trainer\_Id, Trainer\_name FROM Trainer\_Info

WHERE Trainer\_Name LIKE '%o%'

OR TraIner\_Name LIKE '%O%';

8.SELECT \* FROM Module\_Info

WHERE Module\_Name IS NOT null;

**Day 3**

**SQL Functions Assignment**

Exercise.5.1:

|  |
| --- |
|  |

**Problem 1:** Develop a query which will display the module name and module Infra fees of the entire module. The infra fee should be rounded to 2 decimal point.

**Problem 2:** Develop a query which will list all the module id and module names in Module\_Info table where in the first letter should be capital letter.

**Problem 3:** Develop a query which will display the module id and the number of days between the current date and module start date in associate\_status table

**Problem 4:** Develop a query which will concatenate the Module Name and Module id in the following format and display all the modules in the module\_info table.

“< Module Name><Module id>”

**Problem 5:** Develop a query which will display all the Module Name in upper case.

**Problem 6:** Develop a query which will display all the characters between 1 and 3 of the Module name column for all the modules in the Module\_Info table.

**Problem 7**: Develop a query calculate average of all the module base fees, any records whose base fee is null needs to be considered as zero.

**Problem 8:**

Write a query which will convert Trainer\_Info’sTrainer\_Id to Number and add 100000 and display it for all the trainers in the Trainer\_Info table.

**Problem 9:**

Write a query which will convert Base\_Fees into Varchar from the Module\_info table.

And display in the following format

**‘The Base** Fees Amount for the module name’ <Module Name>’ is ’<Base Fees>

**Problem 10:** Write a query which will display the total number of records in Module\_Info table.

**Problem 11**: Develop a query which will give the sum of all base fees of all modules in the Module\_Fees table.

**Problem 12:**  Display the minimum and maximum base fees of the modules.

Ans:

1. SELECT Module\_Name, round(Module\_Infra\_Fee,2)

FROM Module\_Info;

2.SELECT concat(UPPER(SUBSTR(Module\_Id,1,1)),substr(Module\_Id,2,length(Module\_Id))),

concat(UPPER(SUBSTR(Module\_Name,1,1)),substr(Module\_Name,2,length(Module\_Name)))

FROM Module\_Info;

3.SELECT Module\_Id,datediff(CURRENT\_DATE,Start\_Date)

FROM Associate\_Status;

4. SELECT concat('<',Module\_Name,'><',Module\_Id,'>') AS '<Module Name><Module id>'

FROM Module\_Info;

5.SELECT upper(Module\_Name)

FROM Module\_Info;

6.SELECT substr(Module\_Name,1,3)

FROM Module\_Info;

7.SELECT avg(ifnull((Module\_Base\_Fee),0))

FROM Module\_Info;

8.SELECT convert(substr(Trainer\_Id,2,length(Trainer\_Id)),decimal)+100000

FROM Trainer\_Info;

9.ALTER TABLE Module\_Info

MODIFY Module\_Base\_Feevarchar(20);

SELECT concat('The Base Fees Amount for the module name <',Module\_Name,'> is <',ifnull(Module\_Base\_Fee,0),'>')

FROM Module\_Info;

10.SELECT count(\*) FROM Module\_Info;

11.SELECT sum(ifnull(Module\_Base\_Fee,0))

FROM Module\_Info;

12.SELECT max(Module\_Base\_Fee),Min(ifnull(Module\_Base\_Fee,0))

FROM Module\_Info;

**Clauses Assignment**

Exercise6.1

|  |
| --- |
|  |
|  |
|  |  |

Problem Statement:

**Problem 1:**  Develop a SQL query which would retrieve the number of associates enrolled for modules on a specific date grouped by start date and display start date and total number of associates.

**Problem 2:** Develop a SQL query which would retrieve the number of associates enrolled for modules where trainer id is ‘F001’ grouped by start date and display start date and total number of associates.

**Problem 3:** Develop a SQL query which would retrieve the number of associates enrolled for modules where trainer id is ‘F001’ grouped by module start date and display module start date and total number of associates where the total number of associates > 2.

**Problem 4:** Develop a SQL query which displays all the modules in increasing order of module duration.

**Problem 5:** Develop a SQL query which would retrieve and display the associates name, their module enrolled (module name and module id), base fees. Display the records ordering the base fees in descending order.

Ans:

1.SELECT start\_date,count(Associate\_Id)

FROM Associate\_status

GROUP BY start\_date;

2.SELECT start\_date,count(Associate\_Id)

FROM Associate\_status

WHERE Trainer\_Id = 'F001'

GROUP BY start\_date;

3. SELECT start\_date,count(Associate\_Id)

FROM Associate\_status

WHERE Trainer\_Id = 'F001'

GROUP BY start\_date

HAVING count(Associate\_Id) > 2;

4. SELECT \* FROM module\_info

ORDER BY module\_Duration;

5.SELECT Associate\_Id,associate\_status.Module\_Id,Module\_Name,Module\_Base\_Fee

FROM associate\_status INNER JOIN Module\_Info

ON associate\_status.Module\_Id = module\_info.Module\_Id

ORDER BY Module\_Base\_Fee DESC;

**Joins Assignment**

# Exercise 7.1

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Problem Statement:

Write a query to display trainer\_id and batch\_id details in such a way that we get all possible combinations of trainer IDs and batch IDs.

**Ans.7.1**

SELECT Trainer\_Id,Batch\_Id

FROM Trainer\_Info, Batch\_info;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exercise 7.2 | |  | | |
|  | |  |

Write a query to display the records from all columns of table’s associate\_status and batch\_info, wherever the batch\_id in the two tables matches.

**Ans.7.2**

SELECT \*

FROM Associate\_Status INNER JOIN Batch\_Info

ON Associate\_Status.Batch\_Id = Batch\_Info.Batch\_Id;

# 

# Exercise 7.3

Write a query to display the associate IDs of the associates tagged to trainers and all the trainer IDs irrespective of whether there are any associates tagged to them or not.

Hint: Use associate\_status and trainer\_info tables

Note: Use Right Outer Join

**Ans.7.3**

SELECT Associate\_Id,Trainer\_info.Trainer\_Id

FROM Associate\_Status RIGHT JOIN Trainer\_Info

on Associate\_status.Trainer\_Id = Trainer\_Info.Trainer\_Id;

# Exercise 7.4

Write a query to display the associate IDs of the associates tagged to trainers and all the trainer IDs irrespective of whether there are any associates tagged to them or not. Hint: Use associate\_status and trainer\_info tables.

**Ans.7.4**

SELECT Associate\_Id,Trainer\_info.Trainer\_Id

FROM Trainer\_Info LEFT JOIN Associate\_Status

on Associate\_status.Trainer\_Id = Trainer\_Info.Trainer\_Id;

# 

# Exercise 7.5

|  |  |
| --- | --- |
|  |  |

Write a query to display the associate IDs of all the associates and trainer IDs of all trainers irrespective of whether any associate is mapped to a trainer ID and vice versa.

**Hint:** Use associate\_status and trainer\_info tables.

(NOTE: ANSI syntax for Full Outer Join is supported by SQL Server, Not Supported by MySQL, Oracle, and Sybase)

**Prerequisite:** Add an associate\_id which is not mapped to any trainer ID. Take care of the alteration in constraints to the other tables in order to achieve this.

**Ans.7.5**

SELECT Associate\_Id,Trainer\_info.Trainer\_Id

FROM Associate\_Status RIGHT JOIN Trainer\_Info

on Associate\_status.Trainer\_Id = Trainer\_Info.Trainer\_Id

UNION

SELECT Associate\_Id,Trainer\_info.Trainer\_Id

FROM Associate\_Status LEFT JOIN Trainer\_Info

on Associate\_status.Trainer\_Id = Trainer\_Info.Trainer\_Id;

**Sub Queries Assignment**

# Exercise 8.1

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Write a query to display trainer IDs for all associates whose names contain the letter 'i'. Note: Use sub-query to achieve this.

Ans:

SELECT distinct Trainer\_Id FROM Trainer\_Info

WHERE Trainer\_Id in (SELECT Trainer\_Id FROM associate\_status

WHERE Associate\_Id in (SELECT Associate\_Id FROM Associate\_Info

WHERE Associate\_Name LIKE '%i%'));

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exercise 8.2 | |  | | |
|  | |  |

Write a query to display trainer IDs for all associates whose names do not contain the letter 'i'.

**Ans:**

SELECT distinct Trainer\_Id FROM Trainer\_Info

WHERE Trainer\_Id in (SELECT Trainer\_Id FROM associate\_status

WHERE Associate\_Id in (SELECT Associate\_Id FROM Associate\_Info

WHERE Associate\_Name not LIKE '%i%'));

# Exercise 8.3

Write a query to display trainer ID and trainer rating for the trainers who handled J2EE module. Hint: Use trainer\_feedback and associate status tables.

**Ans:**

SELECT Trainer\_Id,Trainer\_Rating

FROM Trainer\_feedback

WHERE Trainer\_Id IN (SELECT Trainer\_IdfROMAssociate\_Status

WHERE Module\_Id = 'J2EE');

# Exercise 8.4

Copy all records of a table into another table using sub-query.

**Prerequisite:** Create a new table trainer\_info\_sabbatical with similar structure as that of trainer\_info. Insert few records into the newly created table (to copy these records into trainer\_info table using subquery)

create the below table

CREATETABLE Trainer\_Info\_Sabbatical

(

Trainer\_Id VARCHAR(20)PRIMARYKEY,

Salutation VARCHAR(7) NOTNULL,

Trainer\_Name VARCHAR(30)NOTNULL,

Trainer\_Location VARCHAR(30)NOTNULL,

Trainer\_Track VARCHAR(15) NOTNULL,

Trainer\_Qualification VARCHAR(100)NOTNULL,

Trainer\_Experiance int,

Trainer\_Email VARCHAR(100) NOTNULL,

Trainer\_Password VARCHAR(20)NOTNULL

);

insert the below records

insertinto Trainer\_Info\_Sabbatical values ('F011','Mr.','Shyju K','Kochi','Java','Bachelor of Technology',9,'shyju@alliance.com','fac11@123');

insertinto Trainer\_Info\_Sabbatical values ('F012','Mr.','Raviraj Kumar','Kochi','Java','Bachelor of Technology',8,'raviarajkumar@alliance.com','fac12@123');

insertinto Trainer\_Info\_Sabbatical values ('F013','Mr.','Suresh Babu N','Mumbai','Testing','Bachelor of Technology',19,'sureshbabun@alliance.com','fac13@123');

**Ans:**

INSERT INTO Trainer\_Info

SELECT \* FROM Trainer\_info\_sabbatical;

# 

# Exercise 8.5

|  |  |
| --- | --- |
|  |  |

Write a query to update trainer locations in trainer\_info table as ‘Kochi’ for all trainers having their experience, figuring in the output of a query which displays all trainer experiences greater than 10 from trainer\_info\_sabbatical table.

**Ans:**

**Exercise 8.6**

|  |  |
| --- | --- |
|  |  |

From trainer\_info table, delete all records where trainer\_experiance matches those in trainer\_info\_sabbatical which are greater than 12.

**Prerequisite :** Have atleast one trainer with experience greater than 12 in both tables.

**Ans:**

DELETE FROM Trainer\_info

WHERE Trainer\_Experiance IN (SELECT Trainer\_Experiance

FROM Trainer\_info\_sabbatical

WHERE Trainer\_Experiance> 12);

# Exercise 8.7

|  |  |
| --- | --- |
|  |  |

Using a corelated subquery display all records from trainer\_info\_sabbatical table which also exist in trainer\_info table

**Prerequisite:** insert atleast one recored into trainer\_info\_sabbatical which is not inserted into trainer\_info table and some records which are common to both.

**Ans:**

SELECT \*

FROM Trainer\_Info\_sabbatical Outer1

WHERE outer1.Trainer\_Id IN ( SELECTTrainer\_Id

FROM Trainer\_Info inner1

WHERE Outer1.Trainer\_Id = inner1.Trainer\_Id);

**Views And Index Assignment**

# Exercise 1

Create a view which displays associate ID and associated trainer ID and batch ID details

**Ans:**

CREATE VIEW view\_Associate\_Status

AS

SELECT Associate\_Id,Trainer\_Id,Batch\_Id

FROM Associate\_Status;

# Exercise 2

Create a new view which displays associate ID and associated trainer ID and batch ID details of ‘b004’ batch.

**Ans:**

CREATE VIEW view\_Associate\_Status1

AS

SELECT Associate\_Id,Trainer\_Id,Batch\_Id

FROM Associate\_Status

WHERE Batch\_Id = 'b004';

# Exercise 3

Drop the view created in Exercise2

**Ans:**

DROP VIEW view\_Associate\_Status1;

# Exercise 4

Create a non-unique index on associate\_name column of associate\_info table.

**Ans:**

CREATE INDEX idx\_associate\_name

ON Associate\_Info(Associate\_Name);

# Exercise 5

Create a unique index on associate\_name column of associate\_info table.

**Ans:**

CREATE UNIQUE INDEX Uidx\_associate\_name

ON Associate\_Info(Associate\_Name);

# Exercise 6

Drop the index created in Exercise 5.

**Ans:**

DROP INDEX Uidx\_associate\_name

ON Associate\_Info;

# Exercise 7

Create a view with:

a)Cascaded check option

b)Local check option

**Ans:**