

SQL HANDSON ASSIGNMENT

Submitted by

ANISHA J

EMPLOYEE ID : 2211630

```
CREATE DATABASE sql_handson_exercises;
```

```
USE sql_handson_exercises;
```

/*EXERCISE 1*/

```
CREATE TABLE Trainer_Info(  
Trainer_Id varchar(20) PRIMARY KEY,  
Salutation varchar(7) ,  
Trainer_Name varchar(30),  
Trainer_Location varchar(30),  
Trainer_Track varchar(15),  
Trainer_Qualification varchar(100),  
Trainer_Experience integer(11),  
Trainer_Email varchar(100),  
Trainer_Password varchar(20));  
DESCRIBE Trainer_Info;
```

```
CREATE TABLE Batch_Info(  
Batch_Id varchar(20) PRIMARY KEY,  
Batch_Owner varchar(30),  
Batch_BU_Name varchar(30));  
DESCRIBE Batch_Info;
```

```
CREATE TABLE Module_Info(  
Module_Id varchar(20) PRIMARY KEY,  
Module_Name varchar(40),  
Module_Duration integer(11));  
DESCRIBE Module_Info;
```

```
CREATE TABLE Associate_Info(  
Associate_Id varchar(20) PRIMARY KEY,  
Salutation varchar(7),  
Associate_Name varchar(30),  
Associate_Location varchar(30),  
Associate_Track varchar(15),  
Associate_Qualification varchar(200),  
Associate_Email varchar(100),  
Associate_Password varchar(20));  
DESCRIBE Associate_Info;
```

```
CREATE TABLE Questions(  
Question_Id varchar(20) PRIMARY KEY,  
Module_Id varchar(20),  
Question_Text varchar(900),  
FOREIGN KEY (Module_Id) REFERENCES Module_Info(Module_Id));  
DESCRIBE Questions;
```

```
CREATE TABLE Associate_Status(  
Associate_Id varchar(20),  
Module_Id varchar(20) ,  
Start_Date varchar(20),  
End_Date varchar(20),  
AFeedbackGiven varchar(20),  
TFeedbackGiven varchar(20),  
FOREIGN KEY(Associate_Id) REFERENCES Associate_Info(Associate_Id),  
FOREIGN KEY(Module_Id) REFERENCES Module_Info(Module_Id));  
DESCRIBE Associate_Status;
```

```
CREATE TABLE Trainer_Feedback(  
Trainer_Id varchar(20) ,  
Question_Id varchar(20),  
Batch_Id varchar(20),  
Module_Id varchar(20),  
Trainer_Rating integer(11),  
FOREIGN KEY(Trainer_Id) REFERENCES Trainer_Info(Trainer_Id),  
FOREIGN KEY(Question_Id) REFERENCES Questions(Question_Id),  
FOREIGN KEY(Batch_Id) REFERENCES Batch_Info(Batch_Id),  
FOREIGN KEY(Module_Id) REFERENCES Module_Info(Module_Id));  
DESCRIBE Trainer_Feedback;
```

```
CREATE TABLE Associate_Feedback(  
Associate_Id varchar(20),  
Question_Id varchar(20),  
Module_Id varchar(20),  
Associate_Rating integer(11),  
FOREIGN KEY(Associate_Id) REFERENCES Associate_Info(Associate_Id),  
FOREIGN KEY(Question_Id) REFERENCES Questions(Question_Id),  
FOREIGN KEY(Module_Id) REFERENCES Module_Info(Module_Id));  
DESCRIBE Associate_Feedback;
```

```
CREATE TABLE Login_Details(  
User_Id varchar(20) PRIMARY KEY,  
User_Password varchar(20));  
DESCRIBE Login_Details;
```

```
ALTER TABLE Associate_Status  
ADD Batch_Id varchar(20) AFTER Module_ID,
```

```
ADD Trainer_Id varchar(20) AFTER Batch_Id,  
ADD FOREIGN KEY (Batch_Id) REFERENCES Batch_Info(Batch_Id),  
ADD FOREIGN KEY (Trainer_Id) REFERENCES Trainer_Info(Trainer_Id);  
DESCRIBE Associate_Status;
```

/*EXERCISE 2*/

```
INSERT INTO Trainer_Info VALUES ('F001','Mr.','PANKAJ GHOSH','Pune','Java','Bachelor  
of Technology',12,'Pankaj.Ghosh@alliance.com','fac1@123'),  
('F002','Mr.','SANJAY RADHAKRISHNAN ','Bangalore','DotNet','Bachelor of  
Technology',12,'Sanjay.Radhakrishnan@alliance.com','fac2@123'),  
('F003','Mr.','VIJAY MATHUR','Chennai','Mainframe','Bachelor of  
Technology',10,'Vijay.Mathur@alliance.com','fac3@123'),  
('F004','Mrs.','NANDINI NAIR','Kolkata','Java','Master of Computer  
Applications',9,'Nandini.Nair@alliance.com','fac4@123'),  
('F005','Miss.','ANITHA PAREKH','Hyderabad','Testing','Master of Computer  
Applications',6,'Anitha.Parekh@alliance.com','fac5@123'),  
('F006','Mr.','MANOJ AGRAWAL','Mumbai','Mainframe','Bachelor of  
Technology',9,'Manoj.Agrawal@alliance.com','fac6@123'),  
('F007','Ms.','MEENA KULKARNI','Coimbatore','Testing','Bachelor of  
Technology',5,'Meena.Kulkarni@alliance.com','fac7@123'),  
('F009','Mr.','SAGAR MENON ','Mumbai','Java','Master of Science In Information  
Technology',12,'Sagar.Menon@alliance.com','fac8@123');  
INSERT INTO Batch_Info VALUES ('B001','MRS.SWATI ROY','MSP'),  
('B002','MRS.ARURNA K','HEALTHCARE'),  
('B003','MR.RAJESH KRISHNAN','LIFE SCIENCES'),  
('B004','MR.SACHIN SHETTY','BFS'),  
('B005','MR.RAMESH PATEL','COMMUNICATION'),  
('B006','MRS.SUSAN CHERIAN','RETAIL & HOSPITALITY'),  
('B007','MRS.SAMPADA JAIN','MSP'),
```

```

('B008','MRS.KAVITA REGE','BPO'),
('B009','MR.RAVI SEJPAL','MSP');
INSERT INTO Module_Info VALUES
('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),
('DOTNET4','.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);
INSERT INTO Associate_Info VALUES
('A001','Miss.','GAYATHRI NARAYANAN','Gurgaon','Java','Bachelor of
Technology','Gayathri.Narayanan@hp.com','tne1@123'),
('A002','Mrs.','RADHIKA MOHAN','Kerala','Java','Bachelor of Engineering In Information
Technology','Radhika.Mohan@cognizant.com','tne2@123'),
('A003','Mr.','KISHORE SRINIVAS','Chennai','Java','Bachelor of Engineering In
Computers','Kishore.Srinivas@ibm.com','tne3@123'),
('A004','Mr.','ANAND RANGANATHAN','Mumbai','DotNet','Master of Computer

```

Applications','Anand.Ranganathan@finolex.com','tne4@123'),
('A005','Miss.','LEELA MENON','Kerala','Mainframe','Bachelor of Engineering In Information
Technology','Leela.Menon@microsoft.com','tne5@123'),
('A006','Mrs.','ARTI KRISHNAN','Pune','Testing','Master of Computer
Applications','Arti.Krishnan@cognizant.com','tne6@123'),
('A007','Mr.','PRABHAKAR SHUNMUGHAM','Mumbai','Java','Bachelor of
Technology','Prabhakar.Shunmugham@honda.com','tne7@123');

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. refered 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 applicatible 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 participats 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 particpate 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. ');

INSERT INTO Associate_Status(Associate_Id, Module_Id, Batch_Id, Trainer_Id, Start_Date, End_Date)
VALUES

('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');
```

/*EXERCISE 3*/

```
UPDATE Trainer_Info SET Trainer_password ='nn4@123'  
WHERE Trainer_ID='f004';  
Select * from Trainer_Info;
```

/*EXERCISE 4*/

```
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';  
Select * from Associate_Status;
```

/*EXERCISE 5*/

```
SELECT * FROM Trainer_Info  
ORDER BY Trainer_Experience DESC LIMIT 5;
```

/*EXERCISE 6*/

```
SELECT * FROM Login_Details ;  
START TRANSACTION ;  
INSERT INTO Login_Details VALUES ('U001' ,'Admin1@123'),  
('U002','Admin2@123');  
ROLLBACK;  
SELECT * FROM Login_Details;  
/*After performing rollback, the login details gets back to initial stage*/
```

/*EXERCISE 7*/

```
CREATE USER 'anisha'@'localhost' IDENTIFIED BY 'root' ;  
  
GRANT CREATE ON sql_handson_exercises TO 'anisha'@'localhost';  
  
GRANT SELECT ON sql_handson_exercises.Login_Details TO 'anisha'@'localhost';  
  
SELECT * FROM Login_Details ;  
  
START TRANSACTION ;  
  
INSERT INTO Login_Details VALUES ('U001' , 'Admin1@123'),  
('U002', 'Admin2@123');  
  
ROLLBACK;  
  
REVOKE CREATE ON sql_handson_exercises FROM 'anisha'@'localhost';  
  
REVOKE SELECT ON sql_handson_exercises.Login_Details FROM 'anisha'@'localhost';  
  
SELECT * FROM Login_Details ;
```

/*EXERCISE 8*/

```
DROP TABLE Login_Details;
```

/*CONSTRAINTS*/

/*EXERCISE 9*/

```
CREATE TABLE Suppliers(  
supplier_id integer(10) NOT NULL,  
supplier_name varchar(50) NOT NULL,  
address varchar(50),  
city varchar(50),  
state varchar(25),  
zip_code varchar(10));  
  
DESCRIBE Suppliers;
```

/*OPERATORS*/

/*EXERCISE 10*/

```
CREATE TABLE Course_Fees(  
  COURSE_CODE varchar(25) PRIMARY KEY,  
  BASE_FEES integer(11),  
  SPECIAL_FEES integer(11),  
  DISCOUNT integer(11));
```

```
CREATE TABLE Course_Fees_History(  
  COURSE_CODE varchar(25) PRIMARY KEY,  
  BASE_FEES integer(11),  
  SPECIAL_FEES integer(11),  
  CREATED_BY varchar(25),  
  Updated_By varchar(25));
```

```
INSERT INTO Course_Fees VALUES  
( '1',180,100,10),( '2',150,110,10),( '3',160,170,5),( '4',150,100,10),( '6',190,100,40);  
INSERT INTO Course_Fees_History VALUES  
( '1',120,123,'Ram','Ramesh'),( '2',150,110,'Bala','Ram'),  
( '3',160,170,'Bala','Vinu'),( '4',170,235,'Ram','Ram'),( '6',190,100,'Vinod','Vinod' );
```

```
SELECT COURSE_CODE,BASE_FEES,SPECIAL_FEES FROM Course_Fees  
UNION  
SELECT COURSE_CODE,BASE_FEES,SPECIAL_FEES FROM Course_Fees_History;
```

/*EXERCISE 11*/

```
SELECT COUNT(*) FROM Course_Fees;  
SELECT COUNT(DISTINCT(COURSE_CODE)) FROM Course_Fees;  
/*Here both the count is equal. Therefore course code is unique in table course_fees*/
```

```
SELECT COUNT(DISTINCT(BASE_FEES)) FROM Course_Fees;
```

```
/*Here both count is not equal. Therefore base fees is not unique in table course_fees*/
```

```
SELECT COUNT(DISTINCT(SPECIAL_FEES)) FROM Course_Fees;
```

```
/*Here both count is not equal. Therefore special fees is not unique in table course_fees*/
```

```
SELECT COUNT(*) FROM Course_Fees_History;
```

```
SELECT COUNT(DISTINCT(COURSE_CODE)) FROM Course_Fees_History;
```

```
/*Here both count is equal. Therefore course code is unique in table course_fees_history*/
```

```
SELECT COUNT(DISTINCT(BASE_FEES)) FROM Course_Fees_History;
```

```
/*Here both count is equal. Therefore base fees is unique in table course_fees_history.*/
```

```
SELECT COUNT(DISTINCT(SPECIAL_FEES)) FROM Course_Fees_History;
```

```
/*Here both count is equal. Therefore special fees is unique in table course_history.*/
```

```
/*FUNCTIONS*/
```

```
/*EXERCISE 12*/
```

```
CREATE TABLE COURSE_INFO(
```

```
COURSE_CODE varchar(10) PRIMARY KEY,
```

```
COURSE_NAME varchar(20) NOT NULL,
```

```
COURSE_DESCRIPTION varchar(25),
```

```
COURSE_START_DATE date ,
```

```
COURSE_DURATION int (100),
```

```
NO_OF_PARTICIPANTS int(200),
```

```
COURSE_TYPE CHAR(3));
```

```
CREATE TABLE Student_Info(
```

```
STUDENT_ID varchar(10) PRIMARY KEY,
```

```
FIRST_NAME varchar(20),
```

```
LAST_NAME varchar(25),
```

```
ADDRESS varchar(150));
```

```
INSERT INTO Course_Fees VALUES ('7',NULL,600,15),('8',NULL,800,30);
INSERT INTO Course_Fees VALUES ('9',300,800,60),('10',175,500,55);
SELECT * FROM Course_Fees;
SELECT MIN(IFNULL(BASE_FEES,0)),MAX(BASE_FEES) FROM Course_Fees;
```

/*EXERCISE 13 */

```
SELECT * FROM course_fees;
ALTER TABLE Course_Fees
ADD COLUMN Infra_Fees DECIMAL(5,3) NOT NULL;
UPDATE Course_Fees SET Infra_Fees = 42.121 WHERE Course_Code = 1;
UPDATE Course_Fees SET Infra_Fees = 41.235 WHERE Course_Code = 2;
UPDATE Course_Fees SET Infra_Fees = 43.524 WHERE Course_Code = 3;
UPDATE Course_Fees SET Infra_Fees = 49.666 WHERE Course_Code = 4;
UPDATE Course_Fees SET Infra_Fees = 44.887 WHERE Course_Code = 6;
UPDATE Course_Fees SET Infra_Fees = 42.583 WHERE Course_Code = 7;
UPDATE Course_Fees SET Infra_Fees = 41.587 WHERE Course_Code = 8;
UPDATE Course_Fees SET Infra_Fees = 42.588 WHERE Course_Code = 9;
UPDATE Course_Fees SET Infra_Fees = 42.631 WHERE Course_Code = 10;
SELECT AVG(Infra_Fees) FROM Course_Fees;
```

/*EXERCISE 14 */

```
INSERT INTO Course_Info VALUES
('001','C++','Programming course','2022-01-10',5,25,'CLR'),
('002','JAVA','Programming course','2021-02-04',8,30,'EL'),
('003','PYTHON','Programming course','2021-06-09',10,40,'OF'),
('004','DEEP LEARNING','Programming course','2022-05-05',9,29,'CLR'),
('005','MACHINE LEARNING','Programming course','2022-02-02',12,50,'EL'),
('006','R PROGRAMMING','Programming course','2021-03-02',10,60,'OF'),
('007','AWS','Programming course','2022-05-12',12,40,'CLR'),
```

```
('008','AZURE','Programming course','2022-04-04',5,55,'EL'),  
('009','C','Programming course','2020-03-02',4,12,'OF');
```

```
SELECT DATEDIFF(curdate(),COURSE_START_DATE) FROM Course_Info;
```

/*EXERCISE 15*/

```
SELECT CONCAT('<',course_name,'>','<',course_code,'>') FROM course_info;
```

/*EXERCISE 16*/

```
SELECT AVG(IFNULL(BASE_FEES,0)) FROM Course_Fees;
```

/*EXERCISE 17*/

```
ALTER TABLE Course_Info ADD COLUMN Message VARCHAR(30);  
UPDATE Course_Info SET Message = "Class Room" WHERE Course_Type = 'CLR';  
UPDATE Course_Info SET Message = "ELearning" WHERE Course_Type = 'EL';  
UPDATE Course_Info SET Message = "OfflineReading" WHERE Course_Type = 'OF';
```

```
SELECT * FROM course_info;
```

```
SELECT Course_Type, Message FROM Course_Info WHERE Course_Type IN ('CLR','EL','OF');
```

/*CLAUSES*/

/*EXERCISE 18*/

```
SELECT COURSE_START_DATE, SUM(NO_OF_PARTICIPANTS) as TOTAL_NO_OF_PARTICIPANTS  
FROM Course_Info GROUP BY COURSE_START_DATE HAVING Course_Start_Date = '2022-04-04';
```

/*EXERCISE 19*/

```
SELECT COURSE_START_DATE, SUM(NO_OF_PARTICIPANTS) as TOTAL_NO_OF_PARTICIPANTS
FROM course_info WHERE course_type='CLR' GROUP BY COURSE_START_DATE ;
```

/*EXERCISE 20*/

```
SELECT COURSE_START_DATE, SUM(NO_OF_PARTICIPANTS) as TOTAL_NO_OF_PARTICIPANTS
FROM Course_Info WHERE course_type='CLR' GROUP BY COURSE_START_DATE HAVING
SUM(NO_OF_PARTICIPANTS)>10;
```

/*EXERCISE 21*/

```
SELECT Course_Name FROM course_info ORDER BY Course_Duration ASC;
```

/*JOINS*/

/*EXERCISE 22*/

```
INSERT INTO course_info VALUES (167,'JAVASCRIPT','Programming course','2022-06-
04',5,85,'EL','ELearning');
```

```
DESCRIBE Student_info;
```

```
ALTER TABLE Student_info ADD COURSE_CODE varchar(10);
```

```
ALTER TABLE Student_info ADD FOREIGN KEY (COURSE_CODE) REFERENCES
COURSE_INFO(COURSE_CODE);
```

```
INSERT INTO Student_info VALUES ('A01','ANISHA','JAMES','KANYAKUMARI','167'),
('AO2','RAJ','KUMAR','CHENNAI','003');
```

```
SELECT Student_info.STUDENT_ID,Student_info.FIRST_NAME,Student_info.LAST_NAME,
Student_info.COURSE_CODE FROM
Student_info INNER JOIN course_info ON
Student_info.COURSE_CODE =course_info.COURSE_CODE
WHERE Student_info.COURSE_CODE = '167';
```

/*EXERCISE 23*/

```
INSERT INTO Course_Fees VALUES
```

```
('167',200, 400, 100, 15),
```

```
('006',200,600,50,10),
```

```
('007',550,450,900,55);
```

```
SELECT course_fees.discount, course_info.course_description FROM course_fees INNER
```

```
JOIN course_info ON course_info.course_code=course_fees.course_code;
```

/*EXERCISE 24*/

```
SELECT CONCAT(First_Name,'_<',Course_Info.Course_Code,'>') AS STUDENT_RECORD
```

```
FROM Student_Info
```

```
LEFT JOIN Course_Info
```

```
ON Student_Info.Course_Code = Course_Info.Course_Code;
```

```
SELECT CONCAT(First_Name,'_<',Course_Info.Course_Code,'>') AS STUDENT_RECORD
```

```
FROM Student_Info
```

```
RIGHT JOIN Course_Info
```

```
ON Student_Info.Course_Code = Course_Info.Course_Code;
```

/*EXERCISE 25*/

```
SELECT * FROM Course_fees;
```

```
INSERT INTO Course_Info VALUES ('010','CLOUD COMPUTING','Learn Programming','2022-11-11',70,11,'CLR','Class Room'),
```

```
('011','RUBY','Learn Programming','2020-02-11',50,1,'EL','ELearning');
```

```
INSERT INTO Course_fees VALUES ('11',1800,1600,20,40),('12','2400','5700',35,20);
```

```
INSERT INTO student_info VALUES ('A04','RENU','KHAN','KERALA','010'),
```

```
('A06','AJI','NAGUL','BOMBAY','011');
```

```
SELECT DISTINCT student_info.student_id FROM student_info INNER JOIN course_fees
```

```
ON student_info.course_code=course_fees.course_code WHERE
```

```
((base_fees+special_fees+infra_fees)*(1-discount/100))<1500;
```


/*EXERCISE 26*/

```
SELECT DISTINCT student_info.student_id,student_info.first_name FROM student_info  
INNER JOIN course_fees ON student_info.course_code=course_fees.course_code WHERE  
((base_fees+special_fees+infra_fees)*(1-discount/100))<1500;
```