Aniket Sarvaiya : 202112076
Nirbhay Movaliya : 202112093
Dharmil Shah : 202112109
Reniz Shah : 202112126

PROJECT DESCRIPTION

Database Title: Employee Training Database Management System

Scope of Database:

It involves keeping records of trainers, trainees, tasks assigned to trainees, reports of the task performed, various courses offered at the company, training schedule, departments under company.

Description:

● Employee training database management system offers a solution to all the data requirements of the company that it uses to train the individuals that take training in their company.
● The timing of the training and duration of training may differ from company-to-company policy. A lot of other factors of the training also depend on the company policy. Therefore, for the whole duration of the database that is from designing to making queries and implementing it, we would keep a company specific approach and not a general approach.
● This database takes into consideration a lot of details for example Trainer details, which may include the name of the trainer, designation of him/her, years of experience they have, the department under which they work and a lot of other general details.
● Trainee details include the name of the trainee, gender, age, address, and other basic details.
● There are a lot of courses that the company offers to trainees to enhance them individual skills that benefit the company.so a list of course offerings and if the trainees have opted for them, their performance on those courses, and other such details are also kept recorded in the database.

Some sample queries are as follows:

● Retrieve the list of trainees who cleared their training in 2007 and are part of the company at present.

● List down the trainers who are trainer for more than 1 courses offered by the company.

● List the total number of trainees who excelled in their probation period.

● Retrieve all the trainees who are under the particular instructor.

● Display team wise details of the project supervised under "XYZ Shah" trainer.

● List of trainees and their details who completed the training from the company and are not part of the company at present.

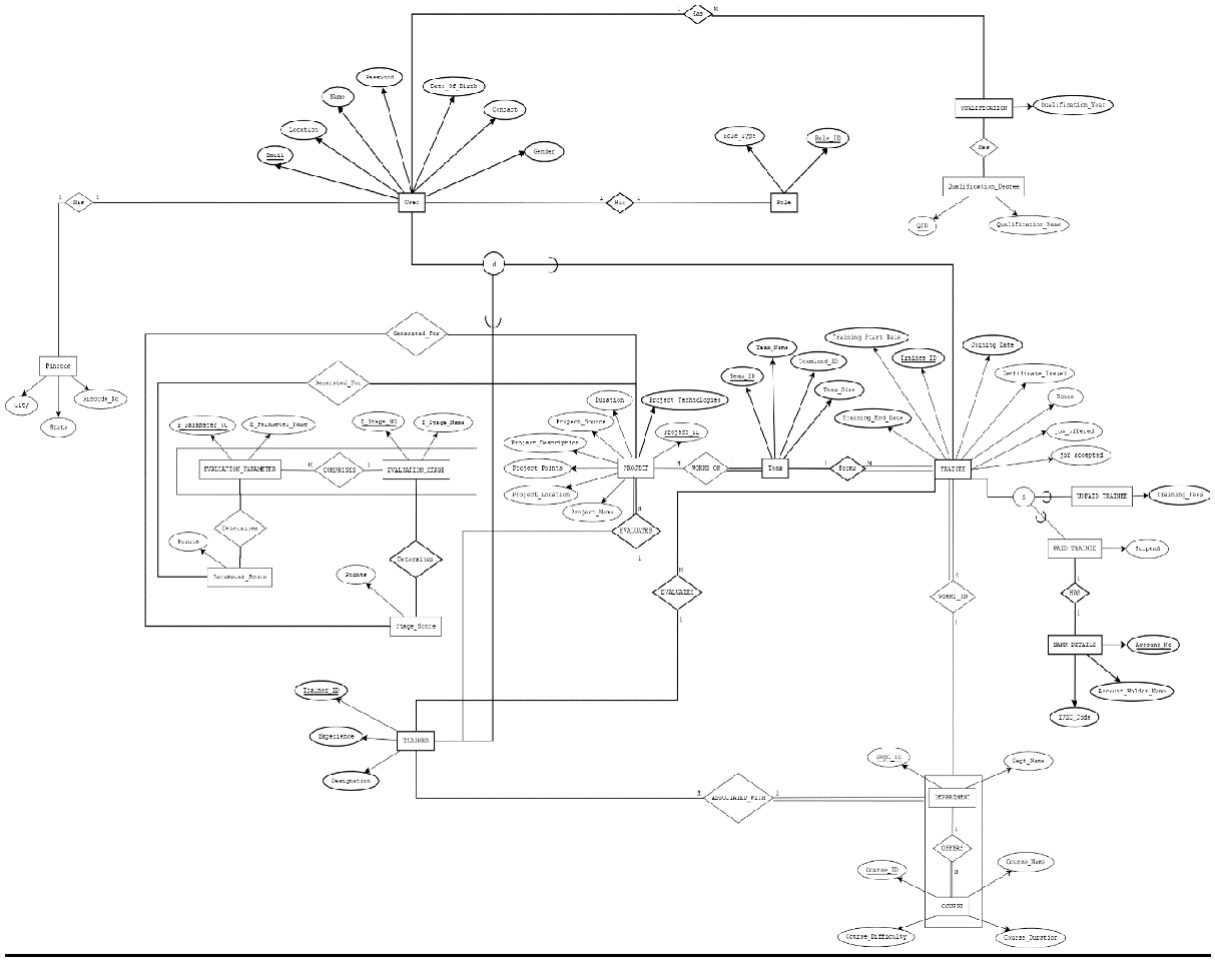
● Display the schedule of training on the basis of batches and also list the count of lecture for particular course per week.

● Count the number of projects given in a particular year to a particular batch and list the year wise best project among all.

● List the instructors and trained trainees having salary more than 10000 per month.

● List the trainees who are already trained and were the part of paid internship.

ER DIAGRAM



FUNCTIONAL DEPENCIES

ROLE

Role_ID -> Role_Type Role_Type -> Role_ID

Prime Attribute-> Role_ID, Role_Name

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, selecting primary key as Role_ID

Minimal Dependency List :

Role_ID -> {Role_Type}

USER

(Names are not unique)

Email -> Name
Email -> Contact
Email -> Date_Of_Birth
Email -> Gender
Email -> Location
Email -> Password
Email -> Pincode_No
Email -> Role_ID

Contact -> Email
Contact -> Name
Contact -> Date_of_Birth
Contact -> Gender
Contact -> Location
Contact –> Password
Contact -> Pincode
Contact -> Role_ID

Pincode -> State
Pincode -> City

Prime Attribute-> Contact, Email

1NF: No, because {Qualification is not atomic attribute}

2NF: No, because {}

3NF: No, because {Pincode -> City, etc}

BCNF: No, because {Pincode -> State, etc }

Therefore, in order to break Non- Atomic Attribute i.e., Qualification we split it into multiple tables as below:

QUALIFICATION

{Email, QID} -> Qualification_Year

Prime Attribute -> Email, QID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, selecting Primary Key -> {Email, QID}.

Minimal Dependency List :

{Email, QID}-> {Qualification_Year}

QUALIFICATION DEGREE

QID-> Qualification_Name Qualification_Name -> QID

Prime Attribute -> QID, Qualification_Name

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, selecting Primary Key -> {QID}.

This would make the table 1NF but not up to BCNF since there are other dependencies like

Pincode -> State etc.

Thus, in order to further normalise it we breakdown the Pincode attribute into separate table.

PINCODE

Pincode_No -> City Pincode_No -> State

Prime Attribute -> Pincode_No

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, selecting Primary Key -> {Pincode_No}.

Thus, In User table selecting Primary Key-> {Email}

Minimal Set of functional dependency:

Email -> {Name, Gender, Contact, Date_Of_Birth, Password, Location, Role_ID, Pincode_No}

TRAINEE

(One trainee can be in only one team)

Trainee_ID -> Joining_Date
Trainee_ID -> Training_Start_Date
Trainee_ID -> Training_End_Date
Trainee_ID -> Score
Trainee_ID -> Certificate_Issued
Trainee_ID -> Dept_ID
Trainee_ID -> Team_ID
Trainee_ID -> Trainer_ID
Trainee_ID -> Score
Trainee_ID -> Job_Offered
Trainee_ID -> Job_Accepted
Trainee_ID -> Email

Prime Attribute-> Trainee_ID

1NF: YES

2NF: YES

3NF: YES

BCNF: YES

Thus, selecting Primary Key {Trainee_ID}

Minimal Dependency List :

{Trainee_ID} -> {Joining_Date, Trainee_Start_Date, Trainee_End_Date, Certificate_Issued, Dept_ID, Team_ID, Trainer_ID, Email, Job_Offered, Job_Accepted, Score}

PAID TRAINEES (This is specialisation of Trainee as reflected from the ER diagram)

Trainee_ID -> Stipend

Prime Attribute -> Trainee_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary Key Selected => Trainee_ID

Minimal Set of Functional Dependency after bringing it into BCNF

Trainee_ID -> {Stipend}

BANK DETAILS

Account_No -> Account_Holder_Name Account_No -> IFSC_Code Account_No -> Trainee_ID

Prime attribute: Account_No

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary key selected => Account_No

Minimal Dependency List:

Account_No -> {Account_Holder_Name, IFSC_Code, Trainee_ID}

UNPAID TRAINEES (This is specialisation of Trainee as reflected from ER diagram)

Trainee_ID -> Training_Fees

Prime Attribute -> Trainee_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary Key Selected => Trainee_ID

Minimal Set of Functional Dependency after bringing it into BCNF

Trainee_ID -> {Training_Fees}

TRAINER

Trainer_ID -> Experience Trainer_ID -> Designation Trainer_ID -> Dept_ID Trainer_ID-> Email

Prime Attribute-> {Trainer_ID}

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary Key Selected => Trainer_ID

Minimal Set of Functional Dependency list :

Trainer_ID -> { Email, Designation, Experience, Dept_ID}

DEPARTMENT

Dept_ID - > Dept_Name

Prime attribute: Dept_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary Key selected => Dept_ID

Minimal Dependency List: Dept_ID -> {Dept_Name}

COURSES

Course_ID -> Course_Name Course_ID -> Course_Duration Course_ID - > Course_Difficulty Course_ID -> Dept_ID

Prime attribute: Course_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary Key selected => Course_ID

Minimal Dependency List:

Course_ID -> {Course_Name, Course_Duration, Course_Difficulty, Dept_ID}

TEAM

Team_ID -> Team_Name Team_ID -> Team_Size Team_ID -> TeamLead_ID

Prime attribute: Team_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, primary key selected => Team_ID

Minimal dependency list:

Team_ID -> {Team_Size, Team_Name,TeamLead_ID}

PROJECT

(Project location may or may not be unique)

Project_ID -> Project_Name Project_ID -> Project_Source Project_ID -> Project_Description Project_ID -> Project_Location Project_ID -> Project_Duration Project_ID -> Project_Technonlogies Project_ID -> Team_ID Project_ID -> Trainer_ID Project_ID -> Project_Points

Prime attribute: Project_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, primary key selected => Project_ID

Minimal dependency list:

Project_ID -> {Project_Name, Project_Source, Project_Description, Project_Location, Project_Duration, Team_ID, Trainer_ID, Project_Points, Project_Technologies}

EVALUATION_STAGE

E_Stage_No -> E_Stage_Name

Prime attribute: E_Stage_No

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary selected => E_Stage_No

Minimal dependency list:

E_Stage_No -> {E_Stage_Name}

STAGE_SCORE

E_Stage_No, Project_ID -> Points

Prime attribute: E_Stage_No, Project_ID

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary selected => {E_Stage_No, Project_ID}

Minimal dependency list:

{E_Stager_No, Project_ID} -> {Points}

EVALUATION_PARAMETER

E_Parameter_No -> E_Parameter_Name E_Parameter_No -> E_Stage_No

Prime attribute: E_Parameter_No

1NF: Yes

2NF: Yes

3NF: Yes

BCNF: Yes

Thus, Primary selected => E_Parameter_No

Minimal dependency list:

{E_Parameter_No} -> {E_Parameter_Name, E_Stage_No}

PARAMETER_SCORE

E_Parameter_No, Project_ID -> Points

Prime attribute: E_Parameter_No, Project_ID

1NF: Yes

2NF: Yes

3NF: Yes

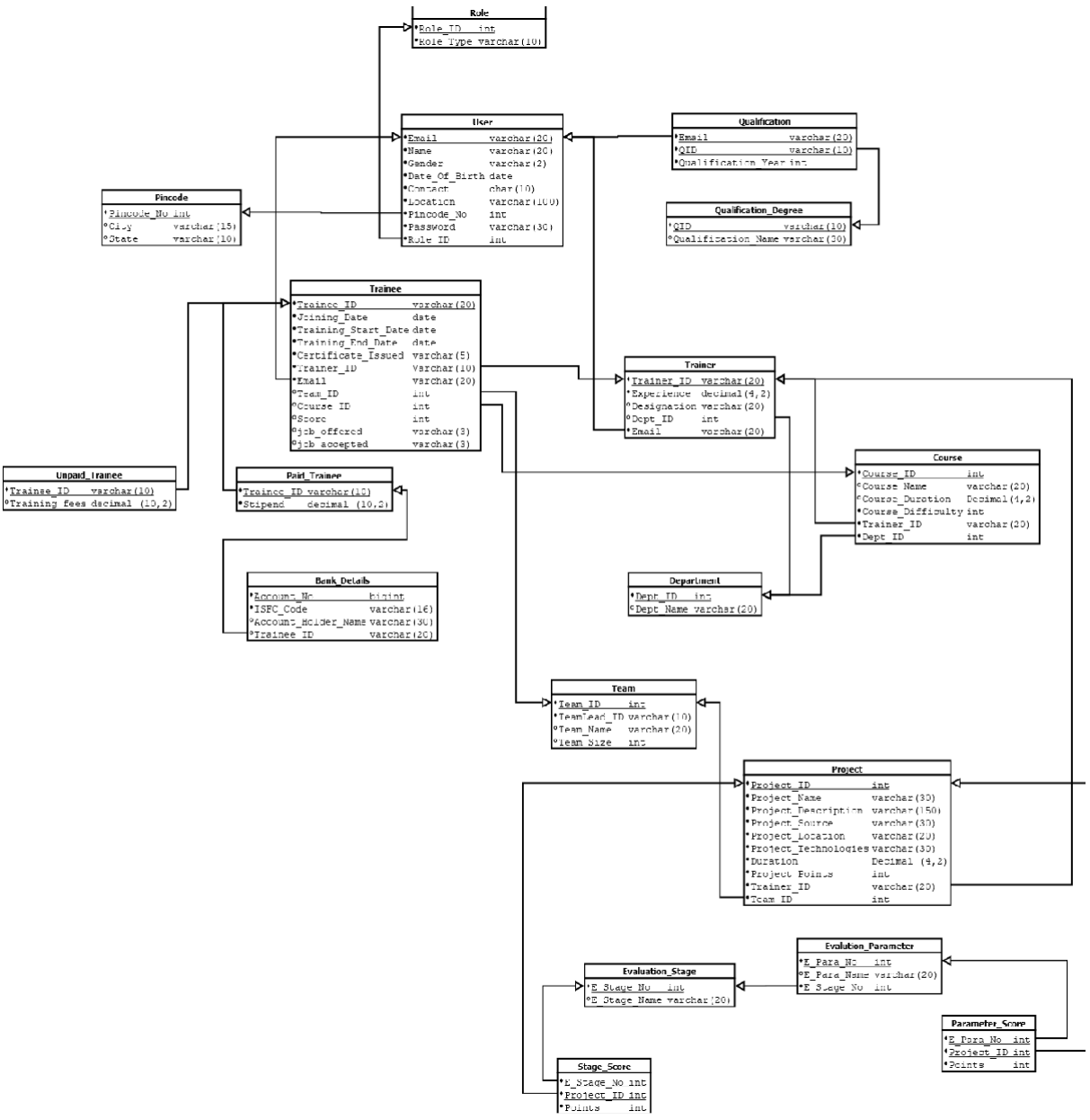
BCNF: Yes

Thus, Primary selected => {E_Parameter_No, Project_ID}

Minimal dependency list:

{E_Parameter_No, Project_ID} -> {Points}

RELATIONAL SCHEMA DIAGRAM



CREATE QUERIES

```
/*SET search_path TO "ETMS ";*/

CREATE TABLE Department(
   Dept_ID SERIAL PRIMARY KEY ,
   Dept_Name varchar(40) NOT NULL
);
CREATE TABLE Pincode(
   Pincode_No int PRIMARY KEY,
   City varchar(20) NOT NULL,
```

```
   State varchar(20) NOT NULL
);
CREATE TABLE Users(
   Email varchar(20) PRIMARY KEY,
   Name varchar(20) NOT NULL,
   Gender varchar(2) NOT NULL,
   Date_Of_Birth date NOT NULL,
   Contact char(10) NOT NULL,
   Location varchar(100) NOT NULL,
   Password varchar(30) NOT NULL, Pincode_No int REFERENCES
Pincode(Pincode_No) ON DELETE CASCADE ON UPDATE CASCADE,
Role_ID int REFERENCES Role(Role_ID) ON DELETE CASCADE ON
UPDATE CASCADE
);
CREATE TABLE Trainer(
   Trainer_ID varchar(20) PRIMARY KEY,
   Experience Decimal(4,2) NOT NULL,
   Designation varchar(20)NOT NULL, Dept_ID int REFERENCES
Department(Dept_ID) ON DELETE CASCADE ON UPDATE CASCADE,
Email varchar(20) REFERENCES Users(Email) ON DELETE CASCADE ON
UPDATE CASCADE
);
CREATE TABLE Course(
   Course_ID SERIAL PRIMARY KEY ,
   Course_Name varchar(70) NOT NULL,
   Course_Duration Decimal(4,2) NOT NULL,
   Course_Difficulty int NOT NULL, Dept_ID int REFERENCES
Department(Dept_ID) ON DELETE CASCADE ON UPDATE CASCADE,
Trainer_ID varchar(20) REFERENCES Trainer(Trainer_ID) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE Team(
   Team_ID SERIAL PRIMARY KEY ,
   Team_Name varchar(20),
   Team_Size int,
   TeamLead_ID varchar(20)
);
CREATE TABLE Trainee(
```

```
    Trainee_ID varchar(20) PRIMARY KEY,
    Joining_Date date NOT NULL,
    Training_Start_Date date NOT NULL,
    Training_End_Date date NOT NULL,
    Certificate_Issues varchar(4) DEFAULT 'NO', Trainer_ID varchar(20)
REFERENCES Trainer(Trainer_ID) ON DELETE CASCADE ON UPDATE
CASCADE,
    Score int,
    Job_offered varchar(3) DEFAULT 'NO',
    Job_accepted varchar(3) DEFAULT 'NO', Email varchar(20)
REFERENCES Users(Email) ON DELETE CASCADE ON UPDATE CASCADE,
Team_ID int REFERENCES Team(Team_ID) ON DELETE CASCADE ON
UPDATE CASCADE,
    Course_ID int REFERENCES Course(Course_ID) ON DELETE CASCADE
ON UPDATE CASCADE
);
CREATE TABLE Paid_Trainee( Trainee_ID varchar(20) REFERENCES
Trainee(Trainee_ID) ON DELETE CASCADE ON UPDATE CASCADE,
    Stipend Decimal(10,2) NOT NULL,
    PRIMARY KEY(Trainee_ID)
);
CREATE TABLE Unpaid_Trainee( Trainee_ID varchar(20) REFERENCES
Trainee(Trainee_ID) ON DELETE CASCADE ON UPDATE CASCADE,
    Training_Fees Decimal(10,2) NOT NULL,
    PRIMARY KEY(Trainee_ID)
);
CREATE TABLE Bank_Details(
    Account_No bigint PRIMARY KEY,
    IFSC_Code varchar(20) NOT NULL,
    Account_Holder_Name varchar(20) NOT NULL,
    Trainee_ID varchar(20) REFERENCES Paid_Trainee(Trainee_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Project(
    Project_ID SERIAL PRIMARY KEY ,
    Project_Name varchar(50) NOT NULL,
    Project_Description text,
    Project_Source varchar(50) NOT NULL,
```

```
   Project_Location varchar(50),
   Project_Technologies varchar(30) NOT NULL,
   Duration Decimal(4,2),
   Project_Points decimal(4,2), Trainer_ID varchar(20) REFERENCES
Trainer(Trainer_ID) ON DELETE CASCADE ON UPDATE CASCADE,
Team_ID int REFERENCES Team(Team_ID) ON DELETE CASCADE ON
UPDATE CASCADE
);
CREATE TABLE Qualification_Degree(
   QID varchar(15) PRIMARY KEY,
   Qualification_Name varchar(80) NOT NULL
);
CREATE TABLE Qualification( QID varchar(15) REFERENCES
Qualification_Degree(QID) ON DELETE CASCADE ON UPDATE CASCADE,
   Qualification_Year int NOT NULL, Email varchar(20) REFERENCES
Users(Email) ON DELETE CASCADE ON UPDATE CASCADE,
   PRIMARY KEY(QID, EMAIL)
);
CREATE TABLE Evaluation_Stage(
   E_Stage_No SERIAL PRIMARY KEY ,
   E_Stage_Name varchar(30) NOT NULL
);
CREATE TABLE Evaluation_Parameter(
   E_Para_No SERIAL PRIMARY KEY ,
   E_Para_Name varchar(30) NOT NULL, E_Stage_No int REFERENCES
Evaluation_Stage(E_Stage_No) ON DELETE CASCADE ON UPDATE
CASCADE
);
CREATE TABLE Parameter_Score(
   E_Para_No int REFERENCES Evaluation_Parameter(E_Para_No) ON
DELETE CASCADE ON UPDATE CASCADE, Project_ID int REFERENCES
Project(Project_ID) ON DELETE CASCADE ON UPDATE CASCADE,
   Points int,
   PRIMARY KEY(E_Para_No, Project_ID)
);
CREATE TABLE Stage_Score( E_Stage_No int REFERENCES
Evaluation_Stage(E_Stage_No) ON DELETE CASCADE ON UPDATE
CASCADE, Project_ID int REFERENCES Project(Project_ID) ON DELETE
```

```
  CASCADE ON UPDATE CASCADE,
    Points decimal(4,2),
    PRIMARY KEY(E_Stage_No, Project_ID)
);
```

INSERT QUERIES

```
  insert into role (Role_Type) values
  ('Trainer'),
  ('Trainee');
  Insert into pincode (Pincode_No, City , State) values
  (364001,'Bhavnagar','Gujarat'),
  (320008,'Ahmedabad','Gujarat'),
  (380004,'Ahmedabad','Gujarat'),
  (380007,'Ahmedabad','Gujarat'),
  (400001,'Mumbai','Maharashtra'),
  (411001, 'Pune','Maharashtra'),
  (395003,'Surat','Gujarat'),
  (302001,'Jaipur','Rajasthan'),
  (600001,'Chennai','Tamil Naidu'),
  (560001,'Bangalore','Karnataka');
  Insert into Users (Email, Name, Gender, Date_Of_Birth, Contact,
  Location, Pincode_No , Password ,Role_ID) values
  ('ram12@gmail.com','Ram','M','1987-05-16',9915941578,'102,HillTop
  Arcade ,Prahlad nagar, Ahmedabad',380007,'15987@',1),
  ('shyam34@gmail.com','Shyam','M','1989-03-29',9985236741,'Block no
  04, MappleCountry,Thaltej Ahmedabad',380004,'shyam12$',1),
  ('dharmil59@gmail.com','Dharmil','M','1999-05-
  02',9987323232,'804,Sapath
  Residency,Bhavnagar',364001,'hello%@12',2),
  ('aniket08@gmail.com','Aniket','F','1991-08-08',7878565652,'A502
  Beverly Hills, Andheri W, Mumbai',400001,'aniket@123',1),
  ('reniz12@gmail.com','Reniz','F','1997-05-
  13',9987656546,'304,Sumeru Park,
  Bhavnagar',364001,'reniz1234@',2),
  ('dhruv123@gmail.com','Dhruv','M','1988-09-
  15',7575753215,'09,Myans Villa,
```

```
Madras,Chennai',600001,'james@123',1),
('nirbhay94@gmail.com','Nirbhay','M','1991-06-
16',7678767830,'605,Arunoday Apartment ,near shan school, Shivaji
Nagar,Pune', 411001,'nirbhay234$',1),
('jay56@gmail.com','Jay','F','1999-05-
15',9898656537,'503,Abhinandan flat, Vaishali
Nagar,Jaipur',302001,'jai@123',2),
('tejas34@gmail.com','Tejas','M','1988-06-21',9879876545,'06,Pavilion
Villas at Brigade Orchards ,Bangalore',560001,'teja#123',1),
('kalpdesai@gmail.com','Kalp','M','1995-08-28',9884860521,' Block no
715, Akshat bungalow near Reliance Mall,Adajan,
Surat',395003,'98765@hello',2),
('jenish@gmail.com','Jenish','M','1985-09-25',9595956219,' 08, Sunrise
Villa, Airport Road, Ahmedabad',380004,'deep159',1),
('nishi@gmail.com','Nishi','F','1997-05-16',9494940123,'A/09,Sukruti
Apartment,Ambavadi,Ahmedabad',380007,'nishu@123',2),
('krupa@gmail.com','Krupa','F','1990-09-08',9393940483,'B/02,Aangi
Residency,Khadki,Pune ',411001,'12345678&kru',1),
('samarth@gmail.com','Samarth','M','1989-03-17',9696964532,'
12,Bhuvan Society,Alandur,Chennai',600001,'sam@123',1),
('shikher@gmail.com','Shikher','M','1998-05-20',9855774578,' D-04
Ankur Apartment,Adajan,Surat',395003,'shik#$%22',2);
Insert into department (dept_name) values
('System Designing'),
('Graphics Designing'),
('FrontEnd'),
('Backend'),
('Database Handling'),
('Application Testing'),
('Technical Support'),
('Cyber Security'),
('Networking');
insert into Trainer (Trainer_ID, Experience , Designation, Dept_ID,
Email) values
('T01',(4.2),'Senior Officer', 1,'ram12@gmail.com'),
('T02',(1.5),'Executive',2,'shyam34@gmail.com'),
('T03',(3.6),'Junior Developer ',3,'aniket08@gmail.com'),
('T04',(2.4),'Junior Developer',4,'dhruv123@gmail.com'),
```

('T05',(4.8),'Senior Executive',5,'nirbhay94@gmail.com'),
('T06',(1.1),'Assistant Developer ',6,'tejas34@gmail.com'),
('T07',(2.9),'Executive',7,'jenish@gmail.com'),
('T08',(6.1),'Senior Developer',8,'krupa@gmail.com'),
('T09',(3.9),'Senior Officer',9,'samarth@gmail.com');
Insert into Course (Course_Name, Course_Duration,
Course_Difficulty,Dept_ID, Trainer_ID) values
('Basics of System Designing',(38.8),1,1,'T01'),
('Intermediate level System Designing',(17),1,1,'T01'),
('Adobe from Intermediate to Expert',(20.2),2,2,'T02'),
('Graphics Designing with LightRoom',(30),3,2,'T02'),
('Basics of React',(25.2),2,3,'T03'),
('React Intermediate to Expert',(15.8),4,3,'T03'),
('Angular Intermediate to Expert',(19.8),4,3,'T03'),
('Java Programming for complete Beginners', (20.50), 4,4,'T04'),
('C# Complete Guide',(30.5),5,4,'T04'),
('.Net Complete Guide',(20.5),3,4,'T04'),
('SQL,PostgreSQL Complete Course',(45.3),3,5,'T05'),
('PostgreSQL Database Management System',(48),3,5,'T05'),
('MySQL Zero to Hero',(25),1,5,'T05'),
('Introduction to FireBase',(37),2,5,'T05'),
('NoSQL with MongoDB',(35),3,5,'T05'),
('Testing with Postman' ,(30.4),3,6, 'T06'),
('Application Testing with Fiddler' ,(30.4),3,6, 'T06'),
('Fundamental of Software Testing',(55.45),3,6,'T06'),
('Technical Support Fundamentals',(18.5),3,7,'T07'),
('The Complete CyberSecurity Course',(40),4,8,'T08'),
('The Complete Networking Fundamentals',(30.8),1,9,'T09');
Insert into Team (Team_Name, Team_Size, TeamLead_ID)
VALUES
('CourseCircuit',3,'E01'),
('StudySharp',3,'E04');
Insert into Trainee (Trainee_Id, Joining_date, Training_Start_Date,
Training_End_Date, Certificate_Issues, Trainer_ID, Email, Team_ID,
Course_ID) values
('E01', '2021-10-15' , '2021-10-18' , '2021-12-10' , 'no' ,'T04'
,'dharmil59@gmail.com' ,'01', 1),
('E02', '2020-11-25' , '2020-11-28' , '2021-02-01' , 'yes' ,'T04'

```
,'reniz12@gmail.com' ,'02',1),
('E03', '2020-03-15' , '2020-03-17' , '2020-04-17' , 'yes' ,'T03'
,'jay56@gmail.com','02',2),
('E04', '2021-10-20' , '2021-10-23' , '2021-12-25' , 'no' ,'T02'
,'kalpdesai@gmail.com','01',3),
('E05', '2021-11-03' , '2021-11-07' , '2022-01-10' , 'no' ,'T02'
,'nishi@gmail.com','01',4),
('E06', '2021-01-10' , '2021-01-15' , '2021-02-03' , 'yes','T03'
,'shikher@gmail.com' ,'02',5);
Insert into Paid_Trainee (Trainee_ID, Stipend) Values
('E01', 6000),
('E02', 5000),
('E05', 4000);
Insert into Unpaid_Trainee (Trainee_ID, Training_Fees) Values
('E03', 6000),
('E04', 8000),
('E06', 11500);
Insert into Bank_Details (Account_No ,IFSC_Code,
Account_Holder_Name,Trainee_ID) Values
(33804587612,'SBIN0020852','Dharmil','E01'),
(75124631689,'SBIN0060318','reniz','E02'),
(82145421156,'SBIN0000301','Nishi','E05');
Insert into Qualification_Degree (QID, Qualification_Name)
values
('BTech IT','Bachelor of Technology in Information Technology'),
('MSC IT','Master of Science in Information Technology'),
('MSC DS','Master of Science in Data Science'),
('MTech AI','Master of Technology in Artificial Intelligence'),
('BCA','Bachelor of Computer Application'),
('MCA','Master of Computer Application'),
('PHD', 'Doctor of Philosophy in Maths'),
('MSC CS', 'Master of Science in Cyber Security'),
('BE IT','Bachelor of Engineering in Information Technology'),
('BE Computer','Bachelor of Engineering in Computer'),
('MTech ICT','Master of Engineering in Information and
Communication Technology '),
('BTech ICT','Bachelor of Engineering in Information and
Communication Technology '),
```

('MEC','Mobile Equipment in Computer');
Insert into Qualification (QID, Qualification_Year, Email)
values
('PHD',2011, 'ram12@gmail.com'),
('MSC IT',2010,'shyam34@gmail.com'),
('BE IT',2021,'dharmil59@gmail.com'),
('MCA',2015,'aniket08@gmail.com'),
('BCA',2019,'reniz12@gmail.com'),
('BTech ICT',2017,'dhruv123@gmail.com'),
('MEC',2013,'nirbhay94@gmail.com'),
('BTech ICT',2021,'jay56@gmail.com'),
('MSC IT',2010,'tejas34@gmail.com'),
('BCA',2017,'kalpdesai@gmail.com'),
('PHD',2007,'jenish@gmail.com'),
('MSC DS',2019,'nishi@gmail.com'),
('BE Computer',2012,'krupa@gmail.com'),
('MTech AI',2011,'samarth@gmail.com'),
('BTech IT',2020,'shikher@gmail.com');
Insert into Project (Project_Name , Project_Description,
Project_Source, Project_Location, Project_Technologies, Duration ,
Trainer_ID, Team_ID)
Values
('Weather Forecasting App' ,' We look at weather data and the future predicted weather to plan our days accordingly. Having visualizations helps us understand that data better' , 'CodeHackers pvt ltd' ,'F:\DAIICT\SEM 1\WeatherForecastingApp', 'React' , 6.5 , 'T02' , 1),
('Android Task Monitoring' ,'This project is exclusively designed to simplify the tracking and monitoring of day-to-day activities of the busy modern life','KND Info ltd' ,'D:\DAIICT\SEM 1\AndroidTaskMonitoring', 'Java' , 4.16 , 'T04', 2),
('Health Tracking App' ,'In this app HeartRate, BloodOxygen, Stress etc are major features which will check users health by their automated sensors by which it will monitor continuously. ' , 'HighTech Data' ,'E:\DAIICT\HealthTrackingApp', 'Java' , 7.5, 'T05' , 1),
('Robotic Arm' ,'In this Project we have used raspberry pi and IOT to connect the robotic arm with mobile app and control it as users command','KND Info ltd' ,null, 'Python' , 6,'T07', null);
Insert into Evaluation_Stage (E_Stage_Name) values

('Project Description'),
('Project System Designing'),
('Project Layout'),
('Implementation'),
('Dissertation'),
('Testing');
Insert into Evaluation_Parameter (E_Para_Name,E_Stage_No) values
('Relevancy',1),
('Practical Usage',1),
('ER Diagram',2),
('Flow Chart',2),
('Use Case Diagram',2),
('Modern UI',3),
('Outcome',3),
('Documented',4),
('Clean Coding',4),
('Research Methodology',5),
('Collection and Analysis',5),
('Discussion and Findings ',5),
('Total time taken',5),
('Robustness of Project',6),
('Scalability',6),
('Bug tracking/defect management',6);
Insert into Parameter_Score(E_Para_No, Project_ID) values
(1,1),
(2,1),
(3,1),
(4,1),
(5,1),
(6,1),
(7,1),
(8,1),
(9,1),
(10,1),
(11,1),
(12,1),
(13,1),
(14,1),

(15,1),
(16,1),
(1,2),
(2,2),
(3,2),
(4,2),
(5,2),
(6,2),
(7,2),
(8,2),
(9,2),
(10,2),
(11,2),
(12,2),
(13,2),
(14,2),
(15,2),
(16,2),
(1,3),
(2,3),
(3,3),
(4,3),
(5,3),
(6,3),
(7,3),
(8,3),
(9,3),
(10,3),
(11,3),
(12,3),
(13,3),
(14,3),
(15,3),
(16,3);
Insert into Stage_Score(E_Stage_No, Project_ID) values
(1,1),
(2,1),
(3,1),

(4,1),
(5,1),
(6,1),
(1,2),
(2,2),
(3,2),
(4,2),
(5,2),
(6,2),
(1,3),
(2,3),
(3,3),
(4,3),
(5,3),
(6,3);

PROJECT QUERIES

1. List down all details of the trainer who mentors in the backend department.

Note: Same details can be fetched for all departments.

=>

```
SELECT *
FROM   users
```

```
    NATURAL JOIN (SELECT *
                  FROM   trainer
                         NATURAL JOIN department
                  WHERE dept_name = 'Backend') AS result
    NATURAL JOIN pincode
    NATURAL JOIN qualification;
```

2. List details of paid trainees who have completed their training in minimum duration.

=>

```
SELECT *
FROM   trainee
    NATURAL JOIN paid_trainee
WHERE training_end_date - training_start_date = (SELECT
    Min(training_end_date
      - training_start_date) AS min_time_for_training
                             FROM   trainee
                             WHERE
        CURRENT_DATE - training_end_date > 0
        AND trainee_id IN (SELECT trainee_id
                           FROM   paid_trainee));
```

3. List department that offers courses but no trainee registered in that course.

=>

```
SELECT *
FROM   department
    natural JOIN (SELECT course_id
                  FROM   course
                  EXCEPT
                  SELECT course_id
                  FROM   trainee)AS result
ORDER  BY dept_id;
```

4. List down count of courses with difficulty 4 on the basis of department.

=>

```
SELECT *
FROM   department
     NATURAL JOIN (SELECT dept_id,
                          Count(course_id)
                   FROM   course
                   WHERE course_difficulty = 4
                   GROUP  BY dept_id)AS r;
```

5. List the difference between total stipend given to paid trainee AND total training fee received from unpaid trainee.

Note: This is very useful information to figure the cost analysis for company

The same difference can also be calculated for employee salary and stipend for paid trainees(But that is out of the scope for this database).

=>

```
SELECT Sum(stipend)           AS total_stipend,
     Sum(training_fees)      AS total_training_fees,
     ( Sum(training_fees) - Sum(stipend) ) AS cost_analysis
FROM   paid_trainee pt
     full JOIN unpaid_trainee ut
          ON ut.trainee_id = pt.trainee_id;
```

6. Set points for Use Case Parameter for project id 2.

Note: Done by trainer to evaluate project based on parameter
Similar points can be set by trainer for all parameters.

=>

```
UPDATE parameter_score
SET    points = 4
WHERE e_para_no = (SELECT e_para_no
                   FROM   evaluation_parameter
                   WHERE e_para_name = 'Use Case Diagram')
```

```
    AND project_id = 2;
```

7. Set points for evaluation stage(Project System designing) based on their parameter points respectively for project id 2.
Note: we calculate the points for each evaluation stage as the average of all the parameters it consists of.
=>

```
UPDATE stage_score
SET    points = (SELECT Avg(points)
                 FROM   parameter_score
                        natural JOIN evaluation_parameter
                        natural JOIN evaluation_stage
                 WHERE  project_id = 2
                        AND e_stage_name = 'Project System Designing')
WHERE project_id = 2
    AND e_stage_no = (SELECT e_stage_no
                      FROM   evaluation_stage
                      WHERE  e_stage_name = 'Project System Designing');
```

8. Set project score for project id 2 based on the evaluation of stages.
Note: we set the project score as the average of points of all evaluation stages.
=>

```
UPDATE project
SET    project_points = (SELECT Avg(points)
                         FROM   stage_score
                         WHERE  project_id = 2)
WHERE  project_id = 2;
```

9. Set score of trainee with id 'E02'.
Note: a similar score can be given to all the trainees by trainer individually.
=>

```
UPDATE trainee
SET    score = 7
```

```
WHERE  trainee_id = 'E02';
```

10. Show a basic report of trainee by creating a view which includes trainee details,score,project worked on, qualification etc.

=>

```
CREATE view report
AS
  (SELECT name,
          gender,
          date_of_birth,
          contact,
          email,
          joining_date,
          training_start_date,
          training_end_date Certificate_Issues,
          job_offered,
          job_accepted,
          qid,
          qualification_year,
          t.team_id,
          team_name,
          project_id,
          project_name,
          project_description,
          project_source,
          project_technologies,
          project_points,
          score
   FROM   users
          NATURAL JOIN trainee
          NATURAL JOIN qualification
          NATURAL JOIN team t
          JOIN project p
            ON p.team_id = t.team_id
   WHERE  trainee_id = 'E01');
```

11. Offer job to trainee with score more than 9.
=>
```
UPDATE trainee
SET    job_offered = 'YES'
WHERE score >= 9;
```

12. List details of trainees who have worked on React as a project(whose project score >= 2.9(out of 5)) and score(individual trainee score) > 7(out of 10).
=>
```
SELECT *
FROM   trainee
      JOIN (SELECT *
           FROM   project
           WHERE project_technologies = 'React'
                AND project_points >= 2.9) AS r
        ON trainee.team_id = r.team_id
WHERE score > 7;
```

13. List trainee details who have good score in 'total time taken'(evaluation parameter) and have project score > 2.5 and their individual score(Score from trainee table) > 7;
=>
```
SELECT trainee_id,
      project_id,
      project_name,
      project_description,
      project_source,
      project_points,
      e_para_name,
      points,
      score
FROM   project
      NATURAL JOIN (SELECT project_id,
```

```
                e_para_name,
                points
            FROM   parameter_score
                NATURAL JOIN evaluation_parameter
            WHERE points > 3
                AND e_para_name = 'Total time taken') AS r
    NATURAL JOIN team
    JOIN trainee
      ON team.team_id = trainee.team_id
WHERE  project_points > 2.5
    AND score > 7
ORDER  BY points DESC;
```

14. List trainees who have offered letters but not accepted the job.

```
=>
SELECT *
FROM   trainee
WHERE job_offered = 'YES'
    AND job_accepted = 'NO';
```

15. List details of teams working on more than 2 projects.

```
=>
SELECT *
FROM   team
    NATURAL JOIN (SELECT team_id,
                Count(project_id) AS count_of_projects
            FROM   project
            GROUP  BY team_id
            HAVING Count(project_id) >= 2)AS result;
```

16. Show count of trainers who have not been assigned any project.

```
=>
SELECT Count(*) AS no_of_trainers_not_assigned
FROM   trainer t
    LEFT JOIN project p
```

```
        ON p.trainer_id = t.trainer_id
WHERE  t.trainer_id NOT IN (SELECT trainer_id
                    FROM   project);
```

17. List top 3 projects(which have been scored by trainer(out of 5)).

=>

```
SELECT *
FROM   project
WHERE project_points IS NOT NULL
ORDER BY project_points DESC
LIMIT 3;
```

18. List top 2 projects of paid trainees on the basis of evaluation stage scores.

=>

```
SELECT e_stage_no,
    e_stage_name,
    project_id,
    project_name,
    points
FROM   stage_score
    NATURAL JOIN evaluation_stage
    NATURAL JOIN project
WHERE project_id IN (SELECT project_id
                FROM   project
                    NATURAL JOIN team
                WHERE teamlead_id IN (SELECT trainee_id
                                FROM   paid_trainee)
                ORDER BY project_points DESC
                LIMIT 2)
ORDER  BY project_id;
```