PROJECT REPORT

ON

DATABASE FOR LINKING THE HR DEPARTMENT AND THE COLLEGE'S RECRUITMENT CELL

ACADEMIC SESSION

Jan –May 2023

SUBMITTED BY:

CHHAVI DHANKHAR	102103605
KHUSHI BANSAL	102103624
NISHTHA KUMARI	102103604
AARCHI WALIA	102103599

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA, PUNJAB 147004

ABSTRACT

As we move ahead post the pandemic, placements especially in the IT sector are observing a boom due to large number of vacancies and ever expanding opportunities in the same field. At the institutional or college level, the HR departments of various companies aid facilitate recruitment, acting as an intermediary between the companies and the students of a college. The integration of college's recruitment cell with the incoming HR departments of the companies aims to eliminate the problems such as confusion between the college and recruiters with respect to applicant data and problems caused by freelancers sometimes in the overall process. The project aims to provide a seamless medium to the institutes and companies to facilitate smooth recruitment of applicants without setbacks while avoiding problems along the way . The project utilizes SQL and PL/SQL by carefully normalizing tables to procure an effective mechanism while minimizing anomalies.

INTRODUCTION

Database Management System

- A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database systems are designed to manage large bodies of information.

The project is Link between the mid - sized company HR, the college and the freelancer for the purpose of recruitment. This package once developed will help mid-sized companies to manage various details pertaining to its hiring. This package is basically developed for the HRs of the company to make their task easier or we can say this package automate their tasks like maintaining List of Freelancers work for them and the personal details, scheduling the interview for them.

The solution for all problems was to automate the system, automation of the data maintenance would reduce the manpower, will result in accurate data and above all increases the efficiency of the concerned department.

Relational Database Management System

- An RDBMS is a type of database management system (DBMS) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS.

E-R DIAGRAM

Here is a graphical representation that depicts relationships among Student, college, company and the freelancers working for the company for the recruitment purposes.ER Diagram is known as Entity-Relationship Diagram, it is used to analyse to the structure of the Database. It shows relationships between entities and their attributes. An ER Model provides a means of communication.

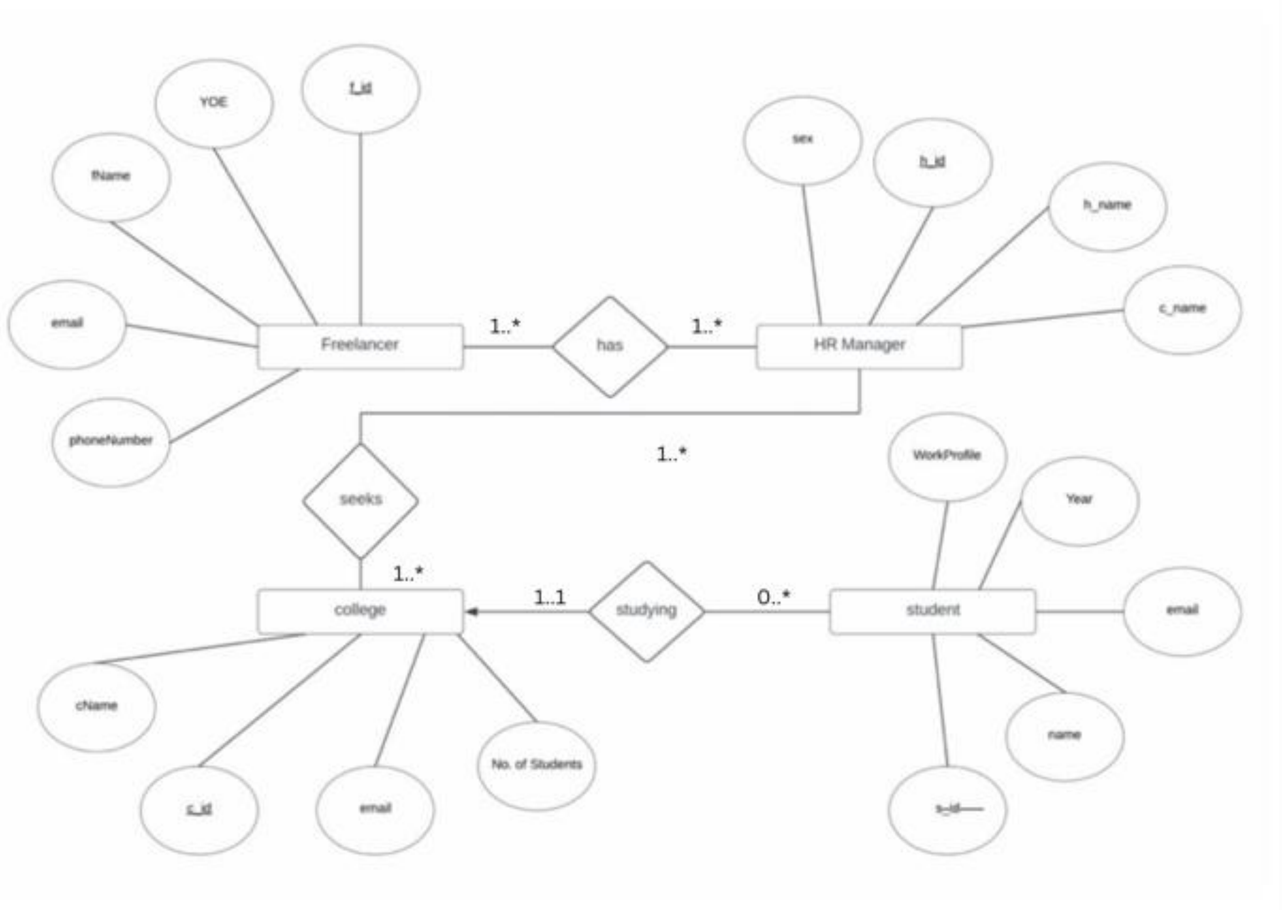


FIG 1. ER diagram for Recruitment portal link between Company freelancer and the college

ER TO RELATIONAL

After designing the ER diagram of the system, we converted it to Relational models which can directly be implemented by any RDBMS. So that it can be easily developed into the database of any of the database system oracle etc.

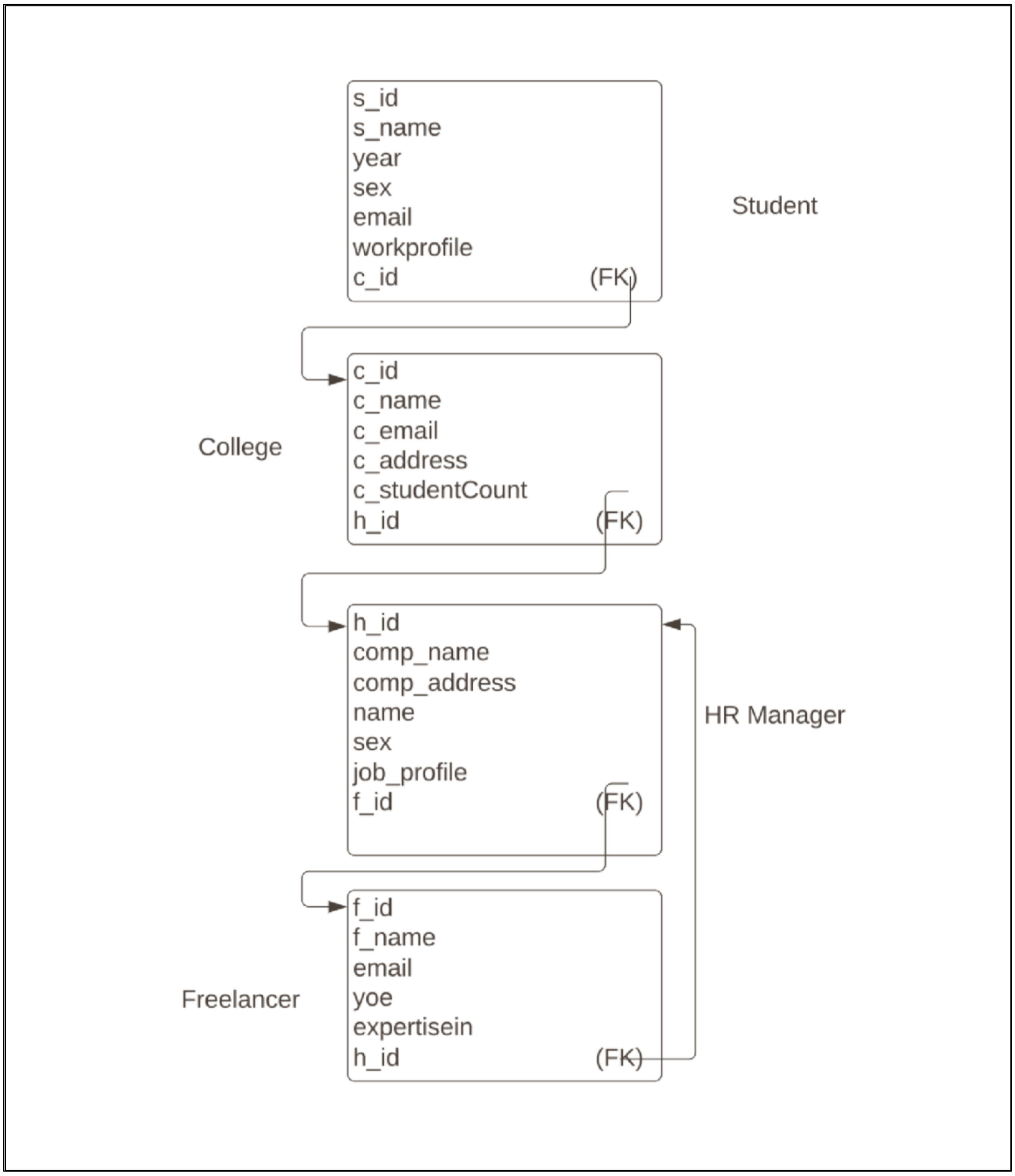


FIG 2 E-R To Relational for Recruitment portal link between Company freelancer and the college

NORMALIZATION

For Removing the different types of anomalies related to insert update delete we can further decomposed out relation database to further more table to remove all the anomalies by using the techniques of different normal form 1NF, 2NF, 3NF, 4NF rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

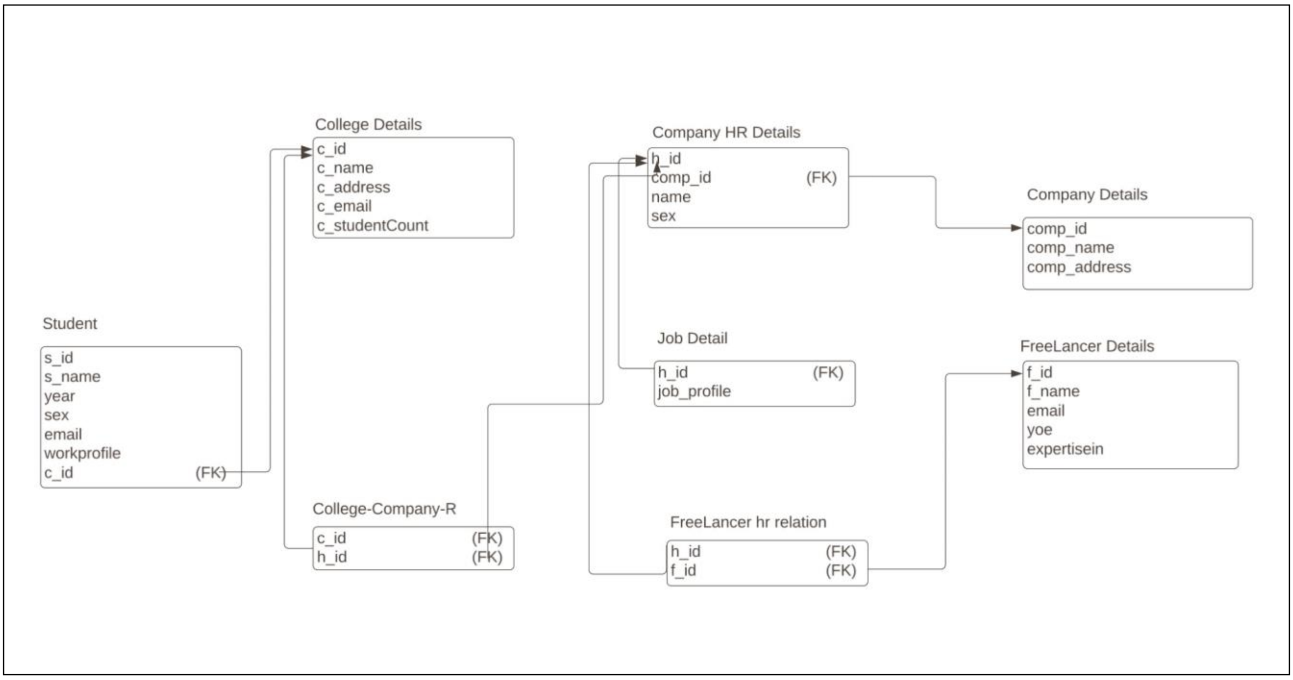


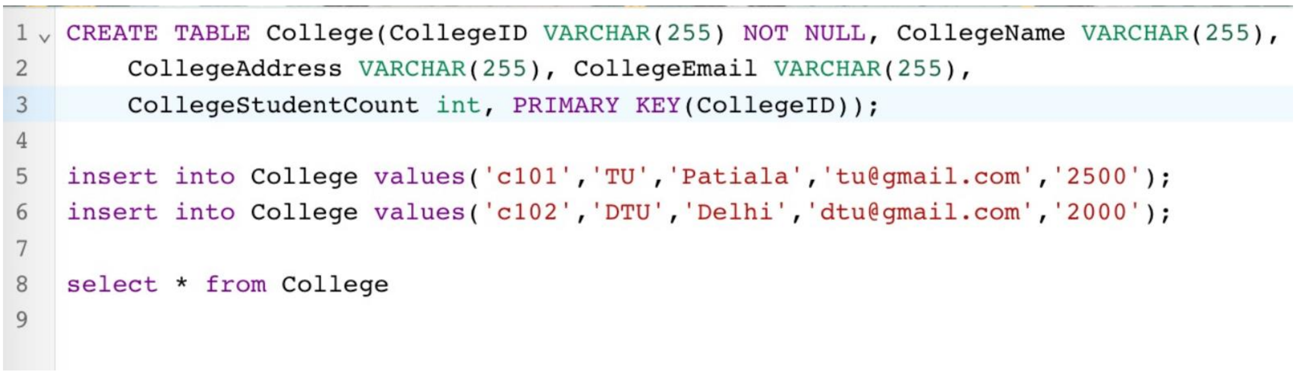
Fig 3. Normalized Relational for Recruitment portal link between Company

CREATION AND INSERTION OF VALUES IN DATABASE

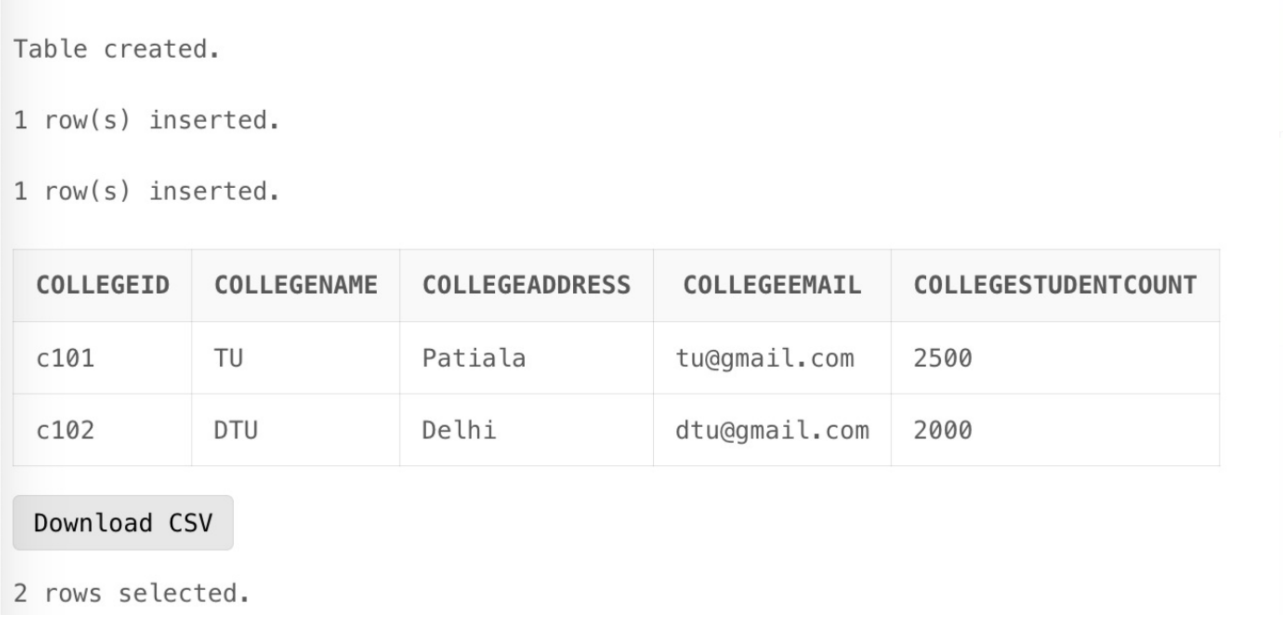
CREATE TABLE College(

CollegeID VARCHAR(255) NOT NULL,
CollegeName VARCHAR(255),
CollegeAddress VARCHAR(255),
CollegeEmail VARCHAR(255),
CollegeStudentCount int,
PRIMARY KEY(CollegeID));
insert into College values('c101','TU','Patiala','tu@gmail.com','500');
insert into College values('c102','DTU','Delhi','dtu@gmail.com','2000');

select * from College



```
CREATE TABLE College(CollegeID VARCHAR(255) NOT NULL, CollegeName VARCHAR(255),
    CollegeAddress VARCHAR(255), CollegeEmail VARCHAR(255),
    CollegeStudentCount int, PRIMARY KEY(CollegeID));

insert into College values('c101','TU','Patiala','tu@gmail.com','2500');
insert into College values('c102','DTU','Delhi','dtu@gmail.com','2000');

select * from College
```



```
Table created.

1 row(s) inserted.

1 row(s) inserted.
```

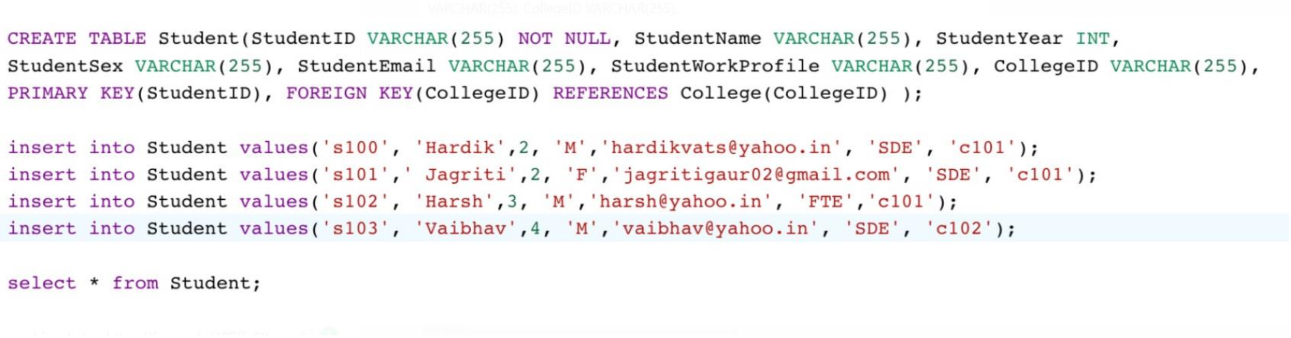
COLLEGEID	COLLEGENAME	COLLEGEADDRESS	COLLEGEEMAIL	COLLEGESTUDENTCOUNT
c101	TU	Patiala	tu@gmail.com	2500
c102	DTU	Delhi	dtu@gmail.com	2000

```
2 rows selected.
```

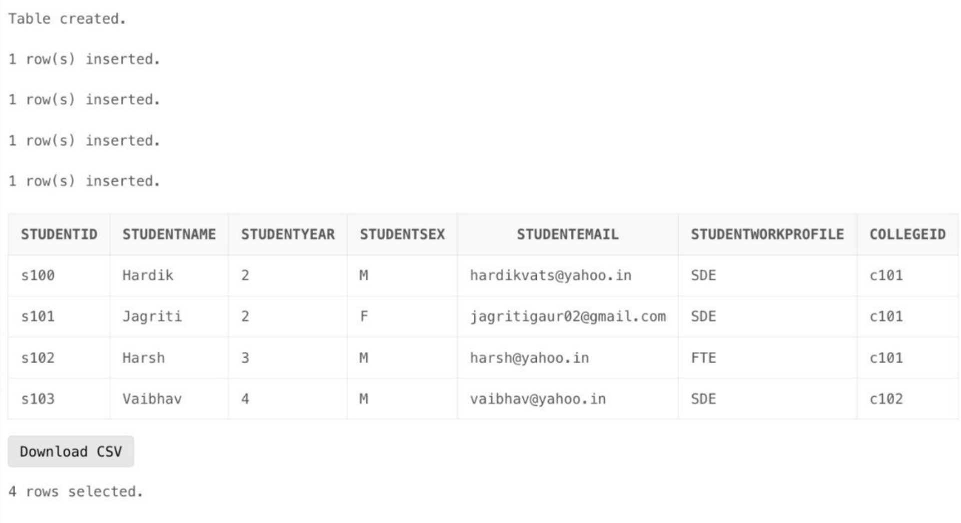
CREATE TABLE Student(StudentID VARCHAR(255) NOT NULL, StudentName VARCHAR(255), StudentYear INT,
StudentSex VARCHAR(255), StudentEmail VARCHAR(255), StudentWorkProfile VARCHAR(255), CollegeID VARCHAR(255),
PRIMARY KEY(StudentID), FOREIGN KEY(CollegeID) REFERENCES College(CollegeID));

insert into Student values('s100', 'Hardik',2, 'M','hardikvats@yahoo.in', 'SDE', 'c101');
insert into Student values('s101',' Jagriti',2, 'F','jagritigaur02@gmail.com', 'SDE', 'c101');
insert into Student values('s102', 'Harsh',3, 'M','harsh@yahoo.in', 'FTE','c101');
insert into Student values('s103', 'Vaibhav',4, 'M','vaibhav@yahoo.in', 'SDE', 'c102');

select * from Student;



```
CREATE TABLE Student(StudentID VARCHAR(255) NOT NULL, StudentName VARCHAR(255), StudentYear INT,
StudentSex VARCHAR(255), StudentEmail VARCHAR(255), StudentWorkProfile VARCHAR(255), CollegeID VARCHAR(255),
PRIMARY KEY(StudentID), FOREIGN KEY(CollegeID) REFERENCES College(CollegeID) );

insert into Student values('s100', 'Hardik',2, 'M','hardikvats@yahoo.in', 'SDE', 'c101');
insert into Student values('s101',' Jagriti',2, 'F','jagritigaur02@gmail.com', 'SDE', 'c101');
insert into Student values('s102', 'Harsh',3, 'M','harsh@yahoo.in', 'FTE','c101');
insert into Student values('s103', 'Vaibhav',4, 'M','vaibhav@yahoo.in', 'SDE', 'c102');

select * from Student;
```

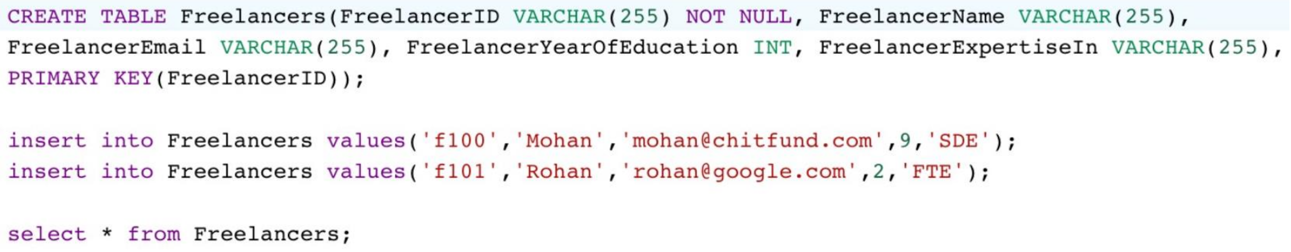


```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

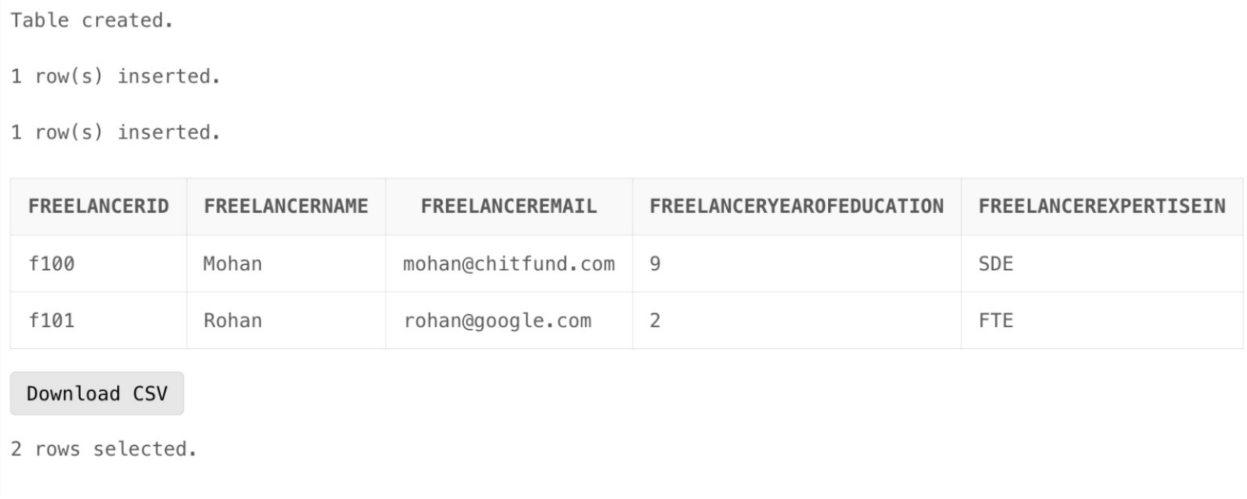
STUDENTID	STUDENTNAME	STUDENTYEAR	STUDENTSEX	STUDENTEMAIL	STUDENTWORKPROFILE	COLLEGEID
s100	Hardik	2	M	hardikvats@yahoo.in	SDE	c101
s101	Jagriti	2	F	jagritigaur02@gmail.com	SDE	c101
s102	Harsh	3	M	harsh@yahoo.in	FTE	c101
s103	Vaibhav	4	M	vaibhav@yahoo.in	SDE	c102

```
4 rows selected.
```

```
CREATE TABLE Freelancers(FreelancerID VARCHAR(255) NOT NULL,
FreelancerName VARCHAR(255),
FreelancerEmail VARCHAR(255), FreelancerYearOfEducation INT,
FreelancerExpertiseIn VARCHAR(255),
PRIMARY KEY(FreelancerID));

insert into Freelancers values('f100','Mohan','mohan@chitfund.com',9,'SDE');
insert into Freelancers values('f101','Rohan','rohan@google.com',2,'FTE');

select * from Freelancers;
```



```
CREATE TABLE Freelancers(FreelancerID VARCHAR(255) NOT NULL, FreelancerName VARCHAR(255),
FreelancerEmail VARCHAR(255), FreelancerYearOfEducation INT, FreelancerExpertiseIn VARCHAR(255),
PRIMARY KEY(FreelancerID));

insert into Freelancers values('f100','Mohan','mohan@chitfund.com',9,'SDE');
insert into Freelancers values('f101','Rohan','rohan@google.com',2,'FTE');

select * from Freelancers;
```

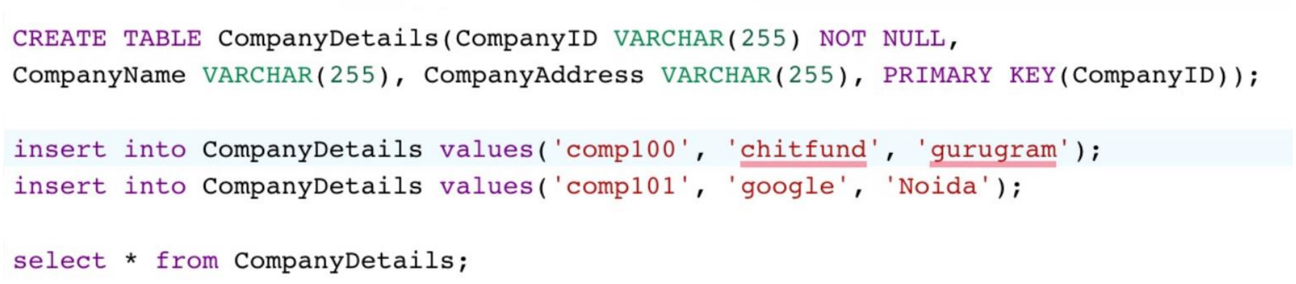


```
Table created.

1 row(s) inserted.

1 row(s) inserted.
```

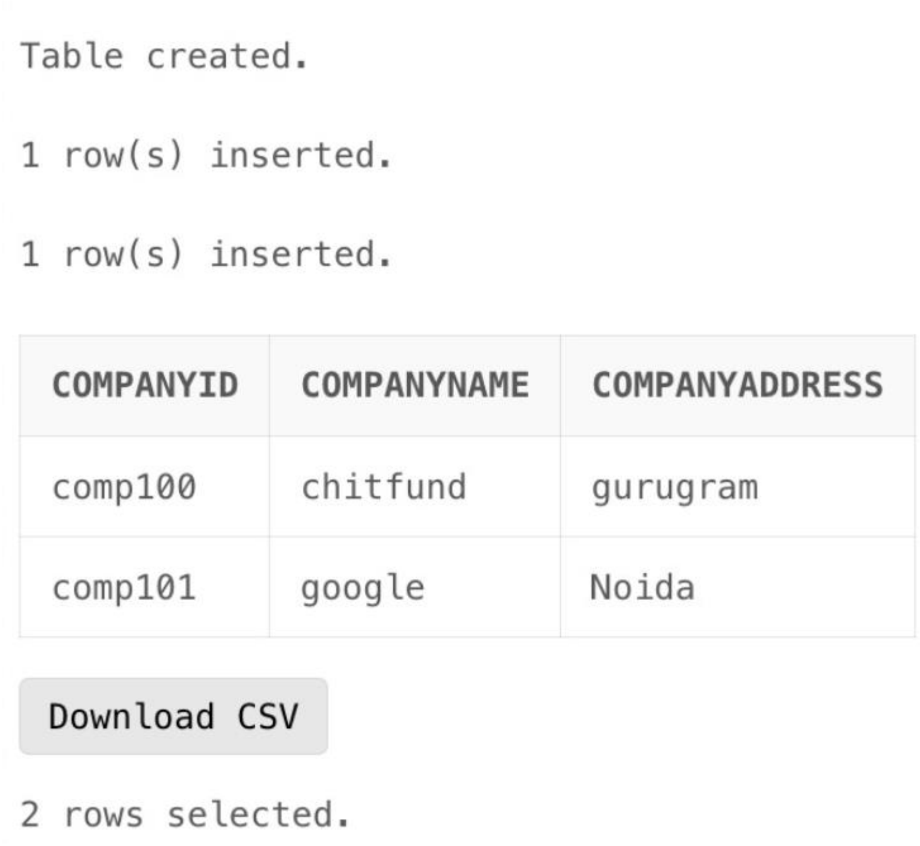
FREELANCERID	FREELANCERNAME	FREELANCEREMAIL	FREELANCERYEAROFEDUCATION	FREELANCEREXPERTISEIN
f100	Mohan	mohan@chitfund.com	9	SDE
f101	Rohan	rohan@google.com	2	FTE

```
2 rows selected.
```

```
CREATE TABLE CompanyDetails(CompanyID VARCHAR(255) NOT NULL,
CompanyName VARCHAR(255), CompanyAddress VARCHAR(255), PRIMARY
KEY(CompanyID));

insert into CompanyDetails values('comp100', 'chitfund', 'gurugram');
insert into CompanyDetails values('comp101', 'google', 'Noida');

select * from CompanyDetails;
```



```
CREATE TABLE CompanyDetails(CompanyID VARCHAR(255) NOT NULL,
CompanyName VARCHAR(255), CompanyAddress VARCHAR(255), PRIMARY KEY(CompanyID));

insert into CompanyDetails values('comp100', 'chitfund', 'gurugram');
insert into CompanyDetails values('comp101', 'google', 'Noida');

select * from CompanyDetails;
```

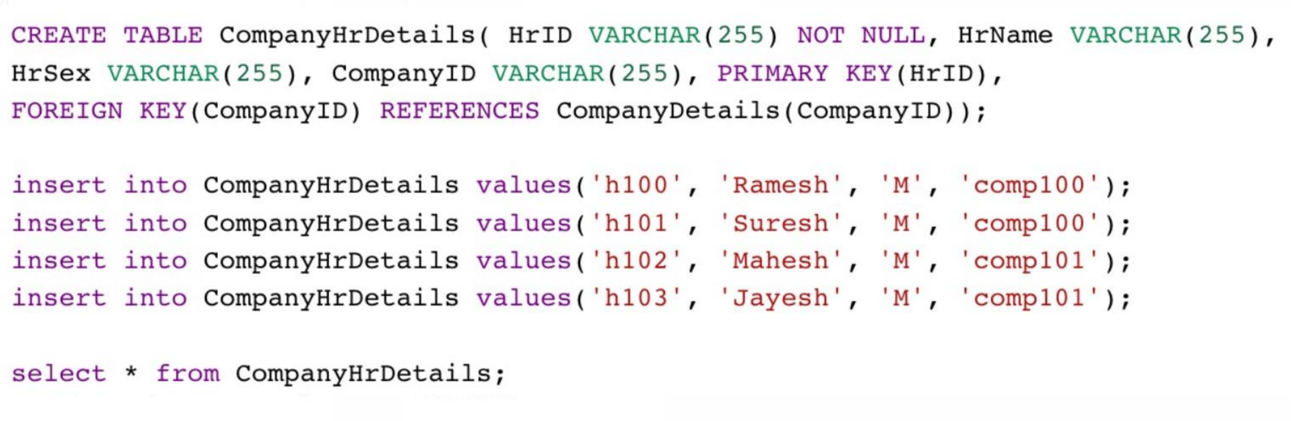


```
Table created.

1 row(s) inserted.

1 row(s) inserted.
```

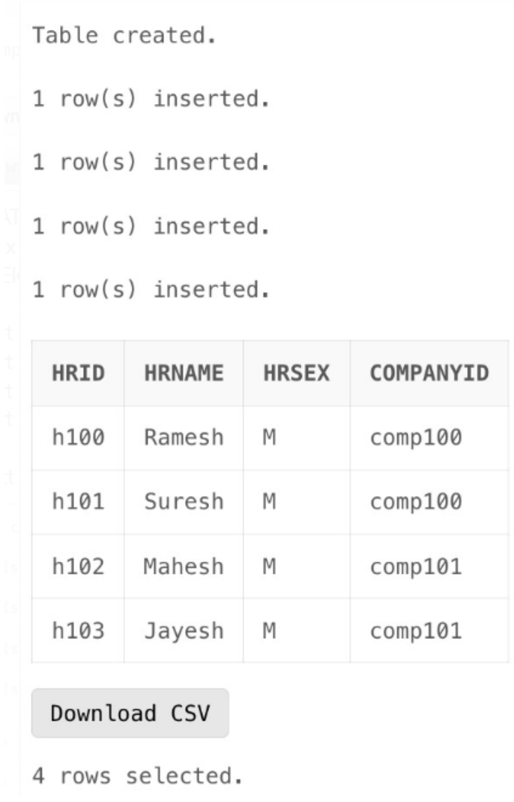
COMPANYID	COMPANYNAME	COMPANYADDRESS
comp100	chitfund	gurugram
comp101	google	Noida

```
2 rows selected.
```

```
CREATE TABLE CompanyHrDetails( HrID VARCHAR(255) NOT NULL, HrName
VARCHAR(255),
HrSex VARCHAR(255), CompanyID VARCHAR(255), PRIMARY KEY(HrID),
FOREIGN KEY(CompanyID) REFERENCES CompanyDetails(CompanyID));

insert into CompanyHrDetails values('h100', 'Ramesh', 'M', 'comp100');
insert into CompanyHrDetails values('h101', 'Suresh', 'M', 'comp100');
insert into CompanyHrDetails values('h102', 'Mahesh', 'M', 'comp101');
insert into CompanyHrDetails values('h103', 'Jayesh', 'M', 'comp101');

select * from CompanyHrDetails;
```



```
CREATE TABLE CompanyHrDetails( HrID VARCHAR(255) NOT NULL, HrName VARCHAR(255),
HrSex VARCHAR(255), CompanyID VARCHAR(255), PRIMARY KEY(HrID),
FOREIGN KEY(CompanyID) REFERENCES CompanyDetails(CompanyID));

insert into CompanyHrDetails values('h100', 'Ramesh', 'M', 'comp100');
insert into CompanyHrDetails values('h101', 'Suresh', 'M', 'comp100');
insert into CompanyHrDetails values('h102', 'Mahesh', 'M', 'comp101');
insert into CompanyHrDetails values('h103', 'Jayesh', 'M', 'comp101');

select * from CompanyHrDetails;
```

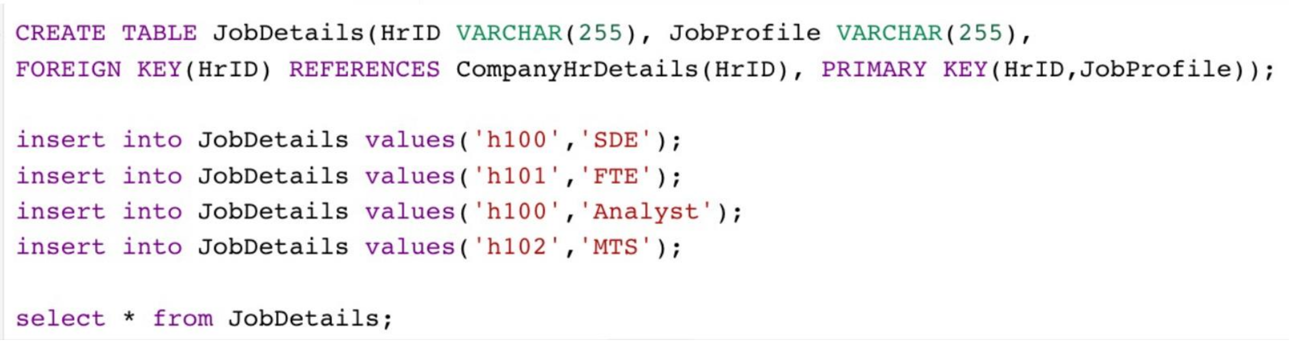


```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

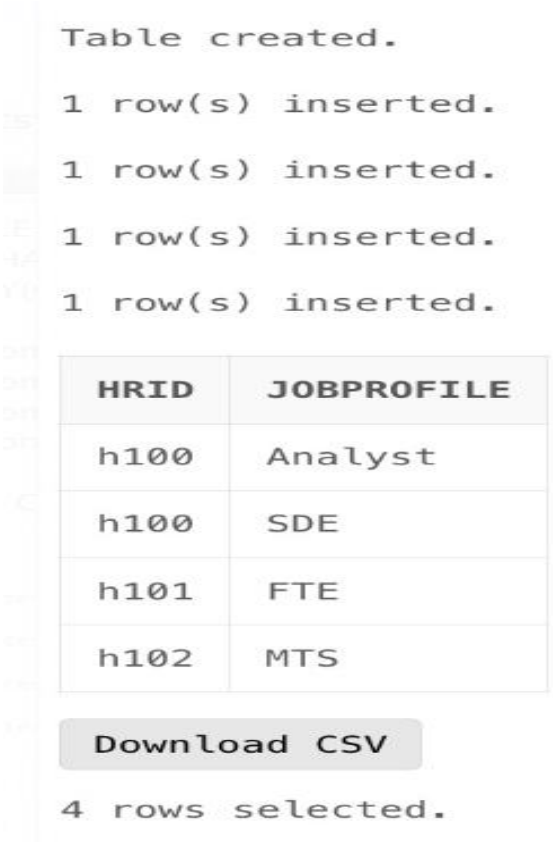
HRID	HRNAME	HRSEX	COMPANYID
h100	Ramesh	M	comp100
h101	Suresh	M	comp100
h102	Mahesh	M	comp101
h103	Jayesh	M	comp101

```
4 rows selected.
```

```
CREATE TABLE JobDetails(HrID VARCHAR(255), JobProfile VARCHAR(255),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID), PRIMARY
KEY(HrID,JobProfile));

insert into JobDetails values('h100','SDE');
insert into JobDetails values('h101','FTE');
insert into JobDetails values('h100','Analyst');
insert into JobDetails values('h102','MTS');

select * from JobDetails;
```



```
CREATE TABLE JobDetails(HrID VARCHAR(255), JobProfile VARCHAR(255),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID), PRIMARY KEY(HrID,JobProfile));

insert into JobDetails values('h100','SDE');
insert into JobDetails values('h101','FTE');
insert into JobDetails values('h100','Analyst');
insert into JobDetails values('h102','MTS');

select * from JobDetails;
```

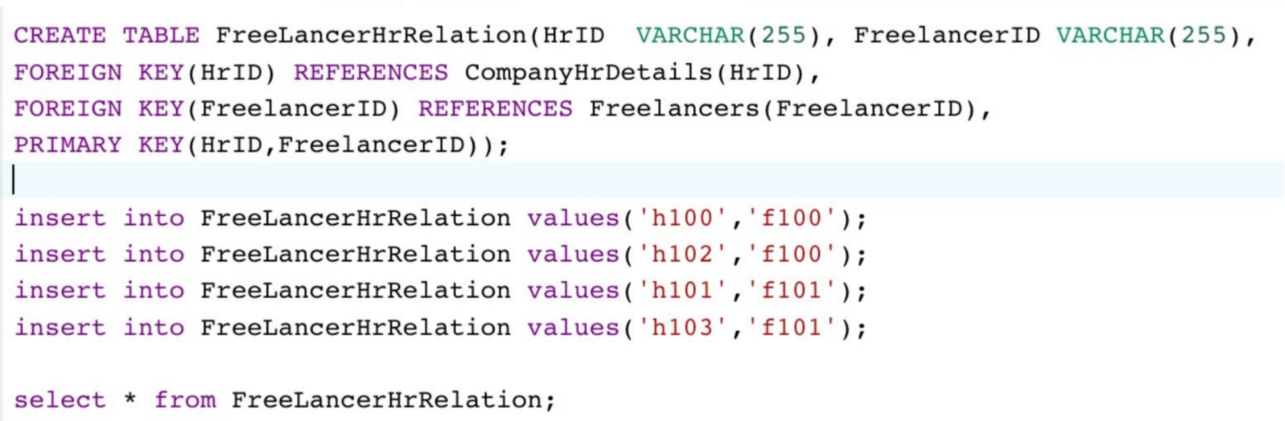


```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

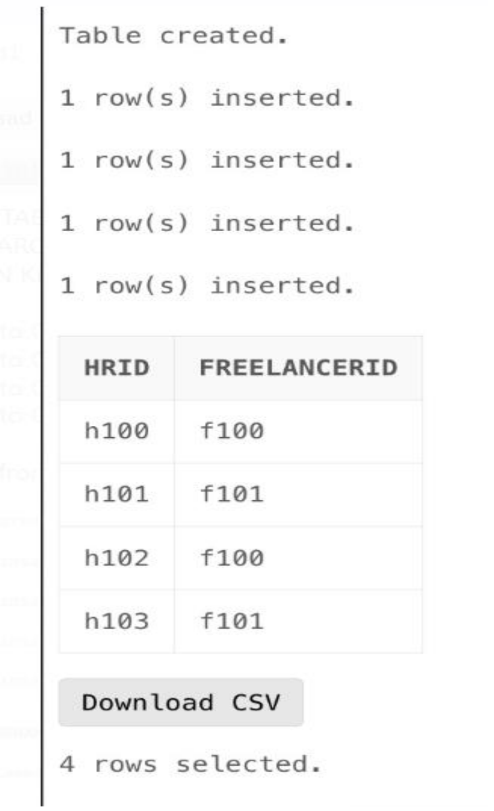
HRID	JOBPROFILE
h100	Analyst
h100	SDE
h101	FTE
h102	MTS

```
4 rows selected.
```

```
CREATE TABLE FreeLancerHrRelation(HrID  VARCHAR(255), FreelancerID
VARCHAR(255),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID),
FOREIGN KEY(FreelancerID) REFERENCES Freelancers(FreelancerID),
PRIMARY KEY(HrID,FreelancerID));

insert into FreeLancerHrRelation values('h100','f100');
insert into FreeLancerHrRelation values('h102','f100');
insert into FreeLancerHrRelation values('h101','f101');
insert into FreeLancerHrRelation values('h103','f101');

select * from FreeLancerHrRelation;
```



```
CREATE TABLE FreeLancerHrRelation(HrID  VARCHAR(255), FreelancerID VARCHAR(255),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID),
FOREIGN KEY(FreelancerID) REFERENCES Freelancers(FreelancerID),
PRIMARY KEY(HrID,FreelancerID));

insert into FreeLancerHrRelation values('h100','f100');
insert into FreeLancerHrRelation values('h102','f100');
insert into FreeLancerHrRelation values('h101','f101');
insert into FreeLancerHrRelation values('h103','f101');

select * from FreeLancerHrRelation;
```

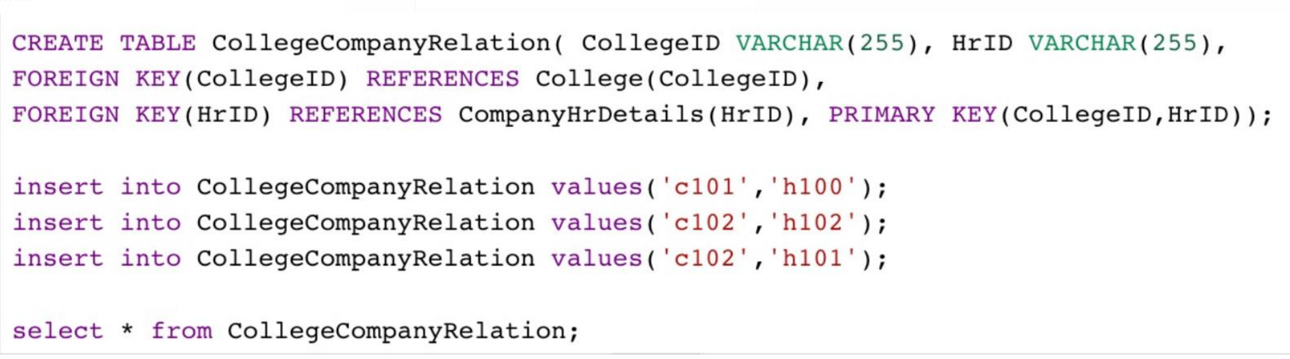


```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

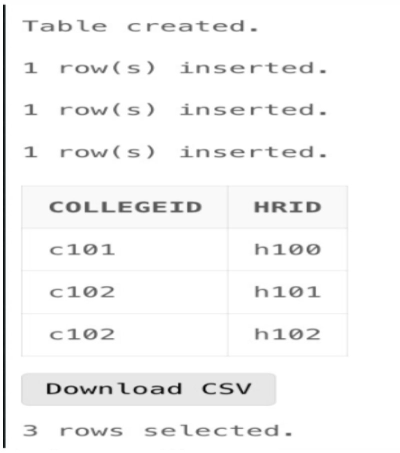
HRID	FREELANCERID
h100	f100
h101	f101
h102	f100
h103	f101

```
4 rows selected.
```

```
CREATE TABLE CollegeCompanyRelation( CollegeID VARCHAR(255), HrID
VARCHAR(255),
FOREIGN KEY(CollegeID) REFERENCES College(CollegeID),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID), PRIMARY
KEY(CollegeID,HrID));

insert into CollegeCompanyRelation values('c101','h100');
insert into CollegeCompanyRelation values('c102','h102');
insert into CollegeCompanyRelation values('c102','h101');

select * from CollegeCompanyRelation;
```



```
CREATE TABLE CollegeCompanyRelation( CollegeID VARCHAR(255), HrID VARCHAR(255),
FOREIGN KEY(CollegeID) REFERENCES College(CollegeID),
FOREIGN KEY(HrID) REFERENCES CompanyHrDetails(HrID), PRIMARY KEY(CollegeID,HrID));

insert into CollegeCompanyRelation values('c101','h100');
insert into CollegeCompanyRelation values('c102','h102');
insert into CollegeCompanyRelation values('c102','h101');

select * from CollegeCompanyRelation;
```



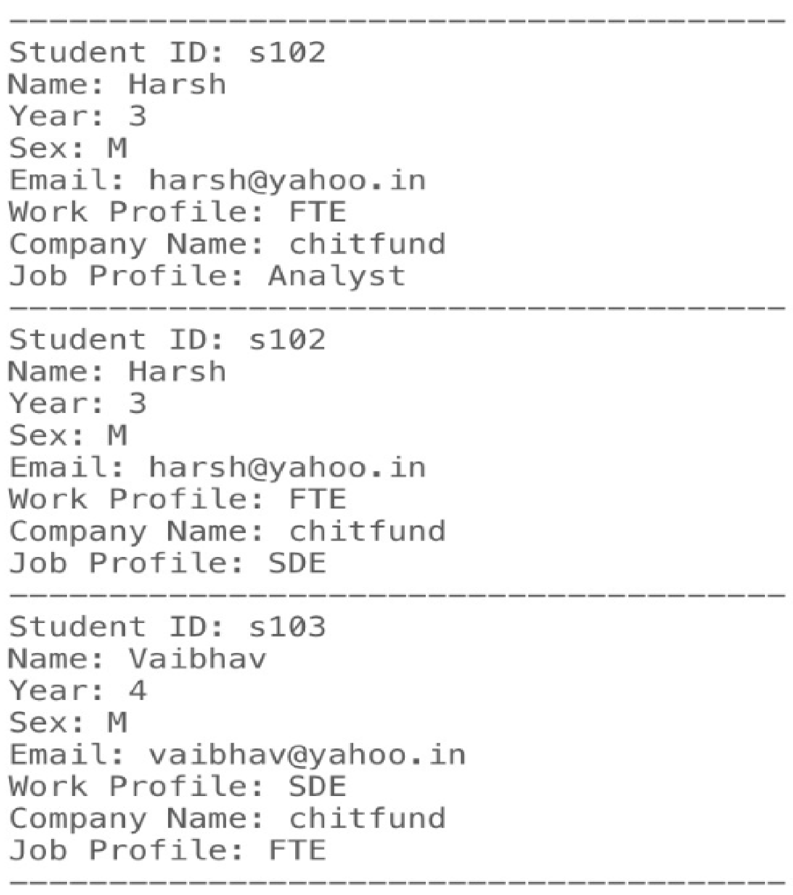
```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

COLLEGEID   HRID
c101        h100
c102        h101
c102        h102
```

```
3 rows selected.
```

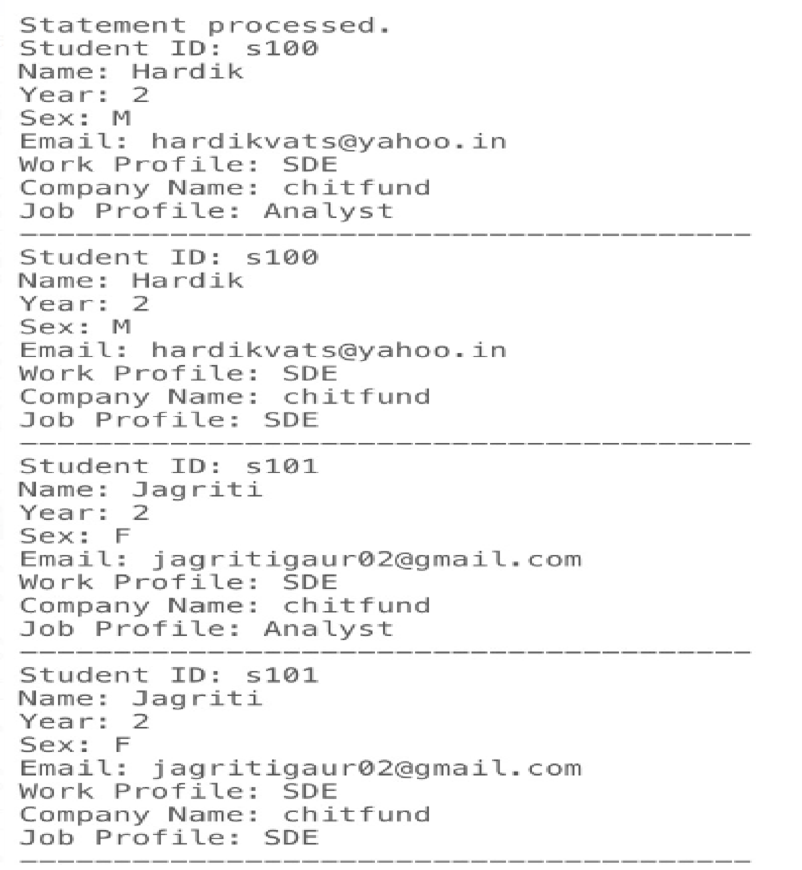
QUERIES

DISPLAYING STUDENT DETAILS

```
DECLARE
    -- Declare a cursor to retrieve student details
    CURSOR c1 IS
        SELECT s.StudentID, s.StudentName, s.StudentYear, s.StudentSex,
s.StudentEmail, s.StudentWorkProfile, c.CompanyName, j.JobProfile
        FROM Student s
        JOIN CollegeCompanyRelation r ON s.CollegeID = r.CollegeID
        JOIN CompanyHrDetails h ON r.HrID = h.HrID
        JOIN CompanyDetails c ON h.CompanyID = c.CompanyID
        JOIN JobDetails j ON h.HrID = j.HrID
        WHERE c.CompanyID = 'comp100'; -- Replace 1 with the desired
company ID

    -- Declare a variable to store retrieved student details
    student_rec c1%ROWTYPE;
BEGIN
    -- Open the cursor and loop through the retrieved records
    OPEN c1;
    LOOP
        FETCH c1 INTO student_rec;
        EXIT WHEN c1%NOTFOUND;

        -- Display the retrieved student details
        DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_rec.StudentID);
        DBMS_OUTPUT.PUT_LINE('Name: ' || student_rec.StudentName);
        DBMS_OUTPUT.PUT_LINE('Year: ' || student_rec.StudentYear);
        DBMS_OUTPUT.PUT_LINE('Sex: ' || student_rec.StudentSex);
        DBMS_OUTPUT.PUT_LINE('Email: ' || student_rec.StudentEmail);
        DBMS_OUTPUT.PUT_LINE('Work Profile: ' ||
student_rec.StudentWorkProfile);
        DBMS_OUTPUT.PUT_LINE('Company Name: ' ||
student_rec.CompanyName);
        DBMS_OUTPUT.PUT_LINE('Job Profile: ' || student_rec.JobProfile);
        DBMS_OUTPUT.PUT_LINE('-------------------------------------------');
    END LOOP;
    CLOSE c1;
EXCEPTION
    -- Handle any exceptions that might occur
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```



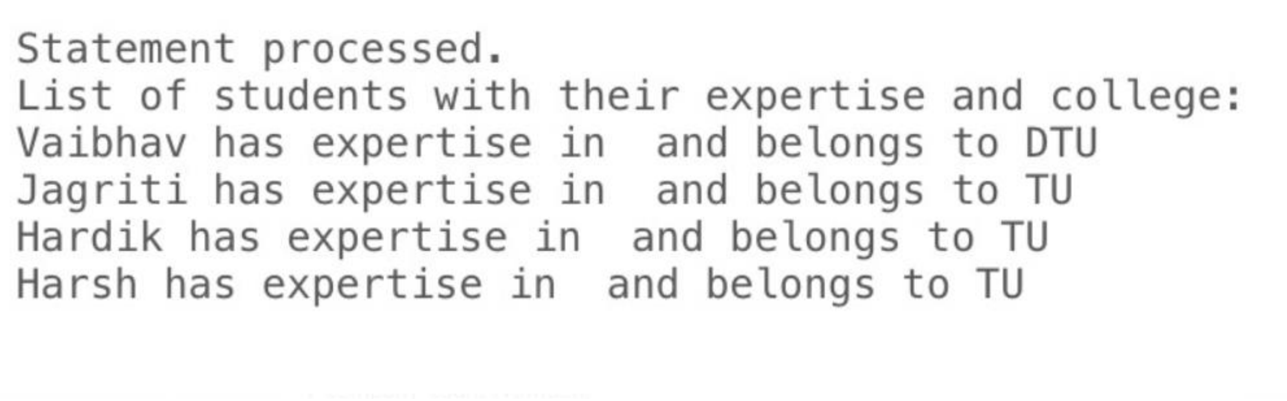
```
---------------------------------------
Student ID: s102
Name: Harsh
Year: 3
Sex: M
Email: harsh@yahoo.in
Work Profile: FTE
Company Name: chitfund
Job Profile: Analyst
---------------------------------------
Student ID: s102
Name: Harsh
Year: 3
Sex: M
Email: harsh@yahoo.in
Work Profile: FTE
Company Name: chitfund
Job Profile: SDE
---------------------------------------
Student ID: s103
Name: Vaibhav
Year: 4
Sex: M
Email: vaibhav@yahoo.in
Work Profile: SDE
Company Name: chitfund
Job Profile: FTE
---------------------------------------
```



```
Statement processed.
Student ID: s100
Name: Hardik
Year: 2
Sex: M
Email: hardikvats@yahoo.in
Work Profile: SDE
Company Name: chitfund
Job Profile: Analyst
---------------------------------------
Student ID: s100
Name: Hardik
Year: 2
Sex: M
Email: hardikvats@yahoo.in
Work Profile: SDE
Company Name: chitfund
Job Profile: SDE
---------------------------------------
Student ID: s101
Name: Jagriti
Year: 2
Sex: F
Email: jagritigaur02@gmail.com
Work Profile: SDE
Company Name: chitfund
Job Profile: Analyst
---------------------------------------
Student ID: s101
Name: Jagriti
Year: 2
Sex: F
Email: jagritigaur02@gmail.com
Work Profile: SDE
Company Name: chitfund
Job Profile: SDE
---------------------------------------
```

LIST OF STUDENTS WITH THEIR EXPERTISE AND COLLEGE

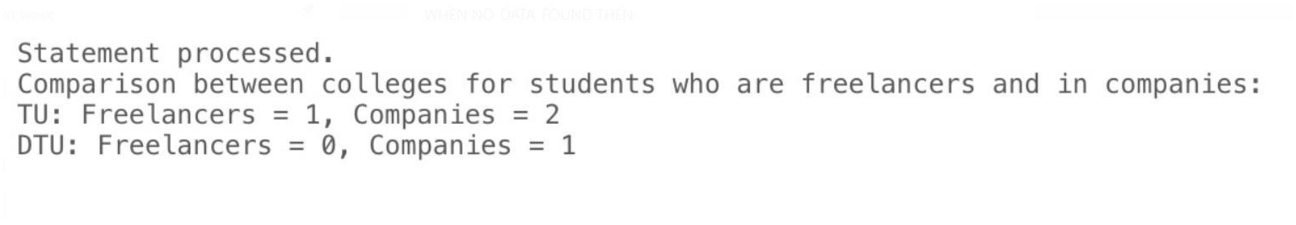
```
DECLARE
  v_college_name College.CollegeName%TYPE;
  v_expertise Freelancers.FreelancerExpertiseIn%TYPE;
  v_student_name Student.StudentName%TYPE;
  CURSOR c2 IS
    SELECT s.StudentName, f.FreelancerExpertiseIn, c.CollegeName
    FROM Student s
    JOIN college c ON s.CollegeID = c.CollegeID
    LEFT JOIN Freelancers f ON s.StudentEmail = f.FreelancerEmail;
BEGIN
  DBMS_OUTPUT.PUT_LINE('List of students with their expertise and college:');
  FOR rec IN c2 LOOP
    v_student_name := rec.StudentName;
    v_expertise := rec.FreelancerExpertiseIn;
    v_college_name := rec.CollegeName;
    DBMS_OUTPUT.PUT_LINE(v_student_name || ' has expertise in ' ||
v_expertise || ' and belongs to ' || v_college_name);
  END LOOP;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' ||
SQLERRM);
END;
```



```
Statement processed.
List of students with their expertise and college:
Vaibhav has expertise in  and belongs to DTU
Jagriti has expertise in  and belongs to TU
Hardik has expertise in  and belongs to TU
Harsh has expertise in  and belongs to TU
```

COMPARISONS BETWEEN COLLEGES FOR STUDENTS WHO ARE FREELANCERS AND IN COMPANY

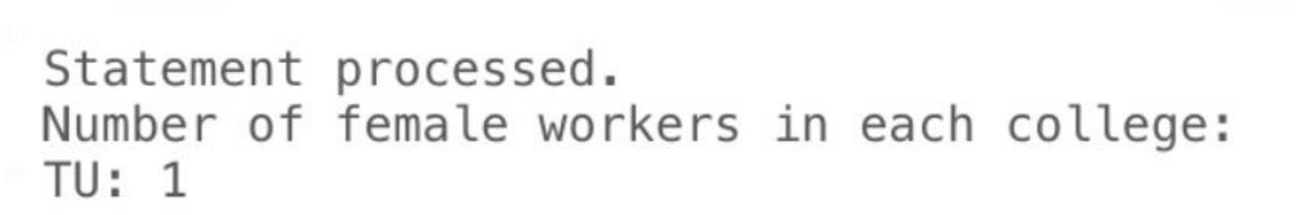
```
 DECLARE
  v_college_name College.CollegeName%TYPE;
  v_freelancer_count NUMBER;
  v_company_count NUMBER;
  CURSOR c3 IS
    SELECT CollegeName, COUNT(CASE WHEN StudentWorkProfile = 'FTE' THEN
1 ELSE NULL END) AS freelancer_count,
           COUNT(CASE WHEN StudentWorkProfile != 'FTE' THEN 1 ELSE NULL
END) AS company_count
    FROM Student
    JOIN College ON Student.CollegeID = College.CollegeID
    GROUP BY CollegeName;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Comparison between colleges for students who are
freelancers and in companies:');
  OPEN c3;
  FETCH c3 INTO v_college_name, v_freelancer_count, v_company_count;
  WHILE c3%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_college_name || ': Freelancers = ' ||
v_freelancer_count || ', Companies = ' || v_company_count);
    FETCH c3 INTO v_college_name, v_freelancer_count, v_company_count;
  END LOOP;
  CLOSE c3;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' ||
SQLERRM);
END;
/
```



```
Statement processed.
Comparison between colleges for students who are freelancers and in companies:
TU: Freelancers = 1, Companies = 2
DTU: Freelancers = 0, Companies = 1
```

NUMBER OF FEMALE WORKERS FROM EACH COLLEGE

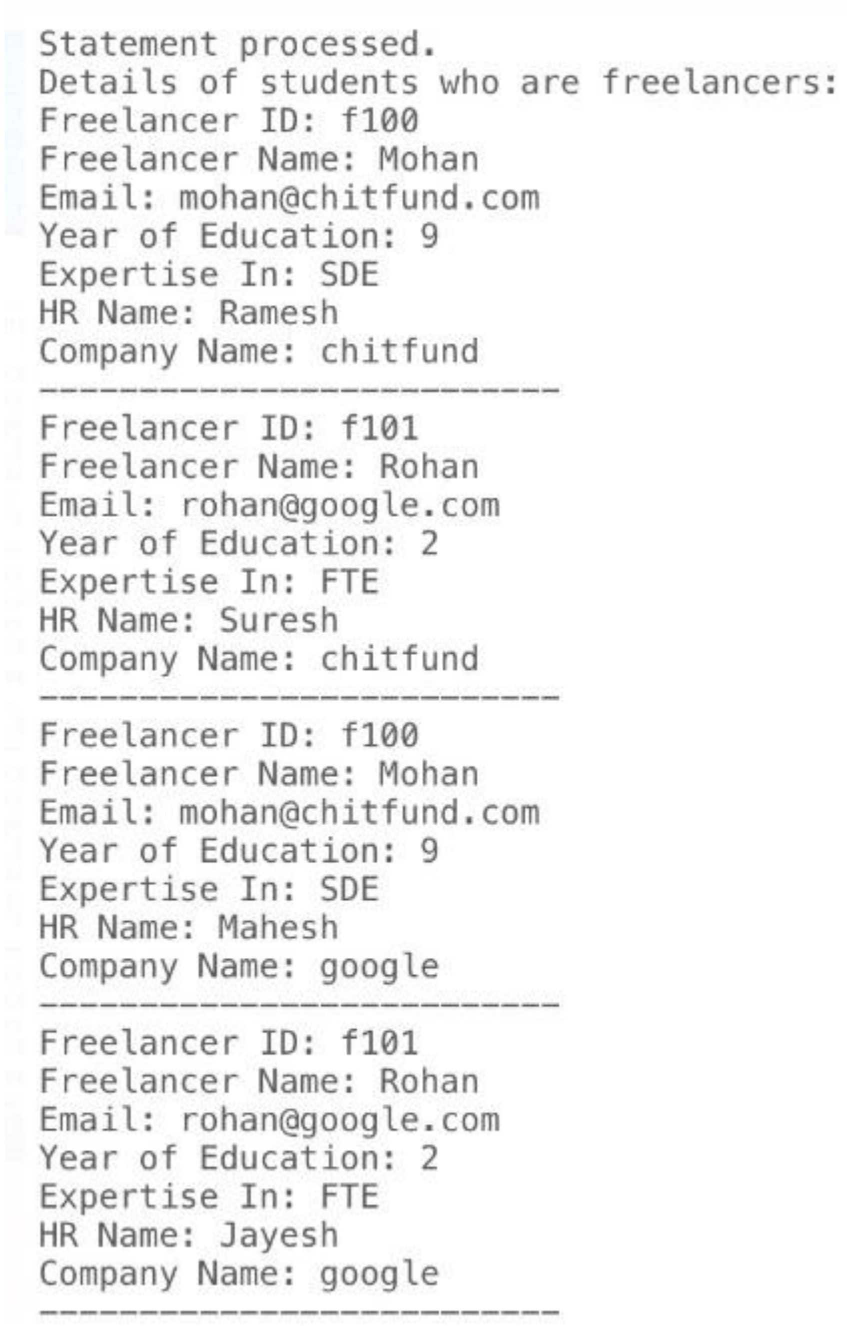
```
 DECLARE
  v_college_name College.CollegeName%TYPE;
  v_female_worker_count NUMBER;
  CURSOR c3 IS
    SELECT CollegeName, COUNT(*) AS female_worker_count
    FROM Student
    JOIN College ON Student.CollegeID = College.CollegeID
    WHERE StudentSex = 'F'
    GROUP BY CollegeName;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Number of female workers in each college:');
  OPEN c3;
  FETCH c3 INTO v_college_name, v_female_worker_count;
  WHILE c3%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE(v_college_name || ': ' ||
v_female_worker_count);
    FETCH c3 INTO v_college_name, v_female_worker_count;
  END LOOP;
  CLOSE c3;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' ||
SQLERRM);
END;
```



```
Statement processed.
Number of female workers in each college:
TU: 1
```

DETAILS OF STUDENTS WHO ARE FREELANCERS

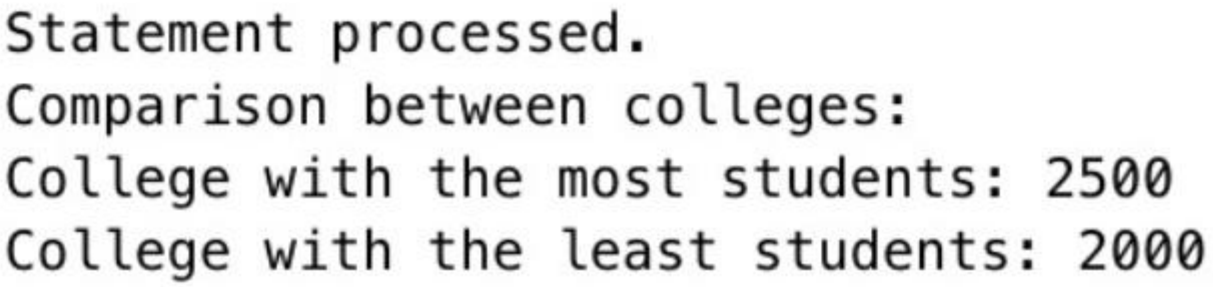
```
DECLARE
  v_freelancer_id Freelancers.FreelancerID%TYPE;
  v_freelancer_name Freelancers.FreelancerName%TYPE;
  v_email Freelancers.FreelancerEmail%TYPE;
  v_year_of_education Freelancers.FreelancerYearOfEducation%TYPE;
  v_expertise Freelancers.FreelancerExpertiseIn%TYPE;
  v_hr_name CompanyHrDetails.HrName%TYPE;
  v_company_name CompanyDetails.CompanyName%TYPE;
  CURSOR c4 IS
    SELECT f.FreelancerID, f.FreelancerName, f.FreelancerEmail,
f.FreelancerYearOfEducation, f.FreelancerExpertiseIn, h.HrName,
d.CompanyName
    FROM Freelancers f
    JOIN FreeLancerHrRelation r ON f.FreelancerID = r.FreelancerID
    JOIN CompanyHrDetails h ON r.HrID = h.HrID
    JOIN CompanyDetails d ON h.CompanyID = d.CompanyID;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Details of students who are freelancers:');
  OPEN c4;
  FETCH c4 INTO v_freelancer_id, v_freelancer_name, v_email,
v_year_of_education, v_expertise, v_hr_name, v_company_name;
  WHILE c4%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE('Freelancer ID: ' || v_freelancer_id);
    DBMS_OUTPUT.PUT_LINE('Freelancer Name: ' || v_freelancer_name);
    DBMS_OUTPUT.PUT_LINE('Email: ' || v_email);
    DBMS_OUTPUT.PUT_LINE('Year of Education: ' || v_year_of_education);
    DBMS_OUTPUT.PUT_LINE('Expertise In: ' || v_expertise);
    DBMS_OUTPUT.PUT_LINE('HR Name: ' || v_hr_name);
    DBMS_OUTPUT.PUT_LINE('Company Name: ' || v_company_name);
    DBMS_OUTPUT.PUT_LINE('--------------------------');
    FETCH c4 INTO v_freelancer_id, v_freelancer_name, v_email,
v_year_of_education, v_expertise, v_hr_name, v_company_name;
  END LOOP;
  CLOSE c4;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' ||
SQLERRM);
END;
```



```
Statement processed.
Details of students who are freelancers:
Freelancer ID: f100
Freelancer Name: Mohan
Email: mohan@chitfund.com
Year of Education: 9
Expertise In: SDE
HR Name: Ramesh
Company Name: chitfund
--------------------------
Freelancer ID: f101
Freelancer Name: Rohan
Email: rohan@google.com
Year of Education: 2
Expertise In: FTE
HR Name: Suresh
Company Name: chitfund
--------------------------
Freelancer ID: f100
Freelancer Name: Mohan
Email: mohan@chitfund.com
Year of Education: 9
Expertise In: SDE
HR Name: Mahesh
Company Name: google
--------------------------
Freelancer ID: f101
Freelancer Name: Rohan
Email: rohan@google.com
Year of Education: 2
Expertise In: FTE
HR Name: Jayesh
Company Name: google
--------------------------
```

COMPARISON OF NUMBER OF STUDENTS IN EACH COLLEGE

```
DECLARE
  v_college_name College.CollegeName%TYPE;
  v_student_count College.CollegeStudentCount%TYPE;
  v_max_count College.CollegeStudentCount%TYPE;
  v_min_count College.CollegeStudentCount%TYPE;
  CURSOR c3 IS
    SELECT CollegeName, CollegeStudentCount
    FROM College;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Comparison between colleges:');
  OPEN c3;
  FETCH c3 INTO v_college_name, v_student_count;
  v_max_count := v_student_count;
  v_min_count := v_student_count;
  WHILE c3%FOUND LOOP
    IF v_student_count > v_max_count THEN
      v_max_count := v_student_count;
    END IF;
    IF v_student_count < v_min_count THEN
      v_min_count := v_student_count;
    END IF;
    FETCH c3 INTO v_college_name, v_student_count;
  END LOOP;
  CLOSE c3;
  DBMS_OUTPUT.PUT_LINE('College with the most students: ' ||
v_max_count);
  DBMS_OUTPUT.PUT_LINE('College with the least students: ' ||
v_min_count);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No records found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLCODE || ' - ' ||
SQLERRM);
END;
```



```
Statement processed.
Comparison between colleges:
College with the most students: 2500
College with the least students: 2000
```