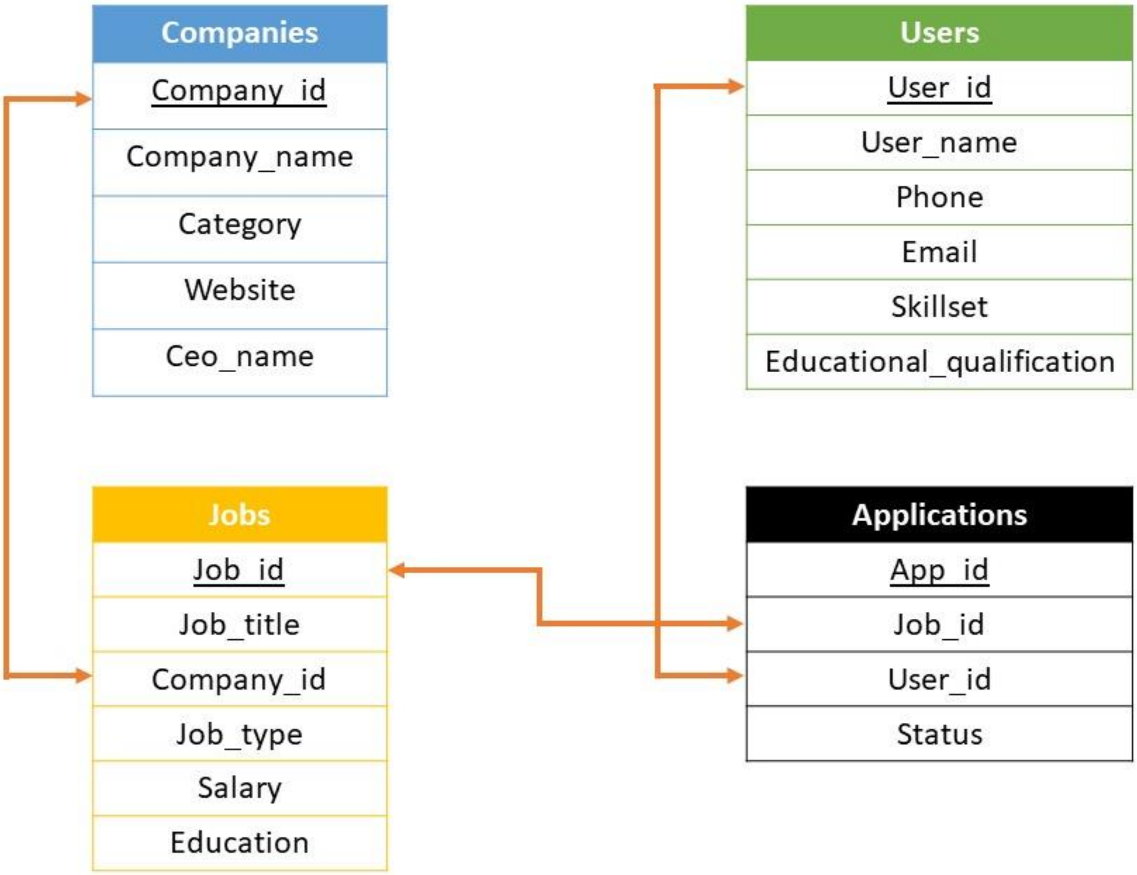

Figure: Schema Diagram

DDL:

```
-- Drop the applications table

DROP TABLE applications;


-- Drop the jobs table

DROP TABLE jobs;


-- Drop the users table

DROP TABLE users;


-- Drop the companies table

DROP TABLE companies;
```

```
-- Create the companies table

CREATE TABLE companies (

   company_id INTEGER NOT NULL,

   company_name VARCHAR(40) NOT NULL,

   category VARCHAR(40),

   website VARCHAR(40),

   ceo_name VARCHAR(40),

   PRIMARY KEY (company_id)

);


-- Create the users table

CREATE TABLE users (

   user_id INTEGER NOT NULL,

   user_name VARCHAR(40) NOT NULL,

   phone NUMBER(15),

   email VARCHAR(40) UNIQUE NOT NULL,

   skillset VARCHAR(40),

   educational_qualification VARCHAR(40),

   PRIMARY KEY (user_id)

);


-- Create the jobs table

CREATE TABLE jobs (

   job_id INTEGER NOT NULL,

   job_title VARCHAR(40) NOT NULL,

   company_id INTEGER NOT NULL,

   job_type VARCHAR(40) CHECK (job_type IN ('Full Time', 'Part Time', 'Contract',
'Temporary', 'Remote')),

   salary NUMBER(20),
   education VARCHAR(40),
   PRIMARY KEY (job_id),
   FOREIGN KEY (company_id) REFERENCES companies(company_id)
   on delete cascade
);

-- Create the applications table
CREATE TABLE applications (
   app_id INTEGER NOT NULL,
   job_id INTEGER NOT NULL,
   user_id INTEGER NOT NULL,
   status VARCHAR(40) CHECK (status IN ('Accepted', 'Rejected', 'Pending')),
   PRIMARY KEY (app_id),
   FOREIGN KEY (job_id) REFERENCES jobs(job_id),
   FOREIGN KEY (user_id) REFERENCES users(user_id)
   on delete cascade
);

-- Add column in the table
ALTER TABLE applications ADD test CHAR(40);

--Modify column definition in the table
alter table applications modify test varchar(40);

--Rename the column name
alter table applications rename column test to test2;

--Drop the column from table
alter table applications drop column test2;
```

DML:

```sql
-- Insert data into the companies table

INSERT INTO companies VALUES(0001, 'Apple', 'Multinational Technology', 'www.apple.com', 'Tim Cook');

INSERT INTO companies VALUES(0002, 'Google', 'Multinational Technology', 'www.google.com', 'Sundar Pichai');

INSERT INTO companies VALUES(0003, 'Electronic Arts', 'Video Game', 'www.ea.com', 'Andrew Wilson');

INSERT INTO companies VALUES(0004, 'Microsoft', 'Multinational Technology', 'www.microsoft.com', 'Sotya Nadella');

INSERT INTO companies VALUES(0005, 'Tesla', 'Multinational Automotive', 'www.tesla.com', 'Elon Mask');

INSERT INTO companies VALUES(0006, 'Samsung', 'Multinational Technology', 'www.samsung.com', 'Kim Hyun-suk');

INSERT INTO companies VALUES(0007, 'Amazon', 'E-commerce', 'www.amazon.com', 'Andy Jassy');

INSERT INTO companies VALUES(0008, 'Sony', 'Multinational Conglomerate', 'www.sony.com', 'Kenichiro Yoshida');

INSERT INTO companies VALUES(0009, 'IBM', 'Multinational Technology', 'www.ibm.com', 'Arvind Krishna');

INSERT INTO companies VALUES(0010, 'Netflix', 'Streaming Media', 'www.netflix.com', 'Reed Hastings');


-- Insert data into the users table

INSERT INTO users VALUES(1001, 'Naimur Rahman', 01751099887, 'naimur.rahman.rs@gmail.com', 'Game Developer', 'Bsc in CSE');

INSERT INTO users VALUES(1002, 'Galib', 01759964341, 'galib@gmail.com', 'Software Developer', 'Bsc in SWE');

INSERT INTO users VALUES(1003, 'Atique Faisal', 01994213076, 'atique@gmail.com', 'Chip Designer', 'Bsc in EEE');

INSERT INTO users VALUES(1004, 'Enan Emon', 01859964299, 'enan@gmail.com', 'Graphics Designer', 'HSC');
INSERT INTO users VALUES(1005, 'Yashrif Arifin', 01776809429, 'yashrif@gmail.com', 'Software Developer', 'Bsc in CSE');

INSERT INTO users VALUES(1006, 'John Smith', 0123456781, 'johnsmith@gmail.com', 'Software Engineer', 'BSc in Computer Science');

INSERT INTO users VALUES(1007, 'Emma Johnson', 0123456782, 'emmajohnson@gmail.com', 'Data Analyst', 'BSc in Mathematics');

INSERT INTO users VALUES(1008, 'Michael Brown', 0123456783, 'michaelbrown@gmail.com', 'Web Developer', 'BSc in Information Technology');

INSERT INTO users VALUES(1009, 'Olivia Davis', 0123456784, 'oliviadavis@gmail.com', 'UI/UX Designer', 'BSc in Multimedia');

INSERT INTO users VALUES(1010, 'Sophia Wilson', 0123456785, 'sophiawilson@gmail.com', 'Data Scientist', 'BSc in Statistics');


-- Insert data into the jobs table

INSERT INTO jobs VALUES(2001, 'Software Developer', 0004, 'Full Time', 20000, 'Bsc in SWE');

INSERT INTO jobs VALUES(2002, 'Game Developer', 0003, 'Part Time', 5000, 'Bsc in CSE');

INSERT INTO jobs VALUES(2003, 'Graphics Designer', 0002, 'Contract', 2000, 'HSC');

INSERT INTO jobs VALUES(2004, 'Software Developer', 0001, 'Part Time', 8000, 'Bsc in CSE');

INSERT INTO jobs VALUES(2005, 'Chip Designer', 0005, 'Full Time', 25000, 'Bsc in EEE');

INSERT INTO jobs VALUES(2006, 'Software Engineer', 0009, 'Full Time', 50000, 'BSc in Computer Science');

INSERT INTO jobs VALUES(2007, 'Data Analyst', 0006, 'Part Time', 25000, 'BSc in Mathematics');

INSERT INTO jobs VALUES(2008, 'Web Developer', 0006, 'Contract', 3000, 'BSc in Information Technology');

INSERT INTO jobs VALUES(2009, 'UI/UX Designer', 0008, 'Full Time', 40000, 'BSc in Multimedia');

INSERT INTO jobs VALUES(2010, 'Data Scientist', 0010, 'Part Time', 35000, 'BSc in Statistics');


-- Insert data into the applications table

INSERT INTO applications VALUES(3001, 2004, 1005, 'Accepted');
```

```sql
INSERT INTO applications VALUES(3002, 2004, 1002, 'Pending');
INSERT INTO applications VALUES(3003, 2001, 1005, 'Rejected');
INSERT INTO applications VALUES(3004, 2001, 1002, 'Accepted');
INSERT INTO applications VALUES(3005, 2005, 1003, 'Accepted');
INSERT INTO applications VALUES(3006, 2003, 1004, 'Pending');
INSERT INTO applications VALUES(3007, 2002, 1001, 'Accepted');
INSERT INTO applications VALUES(3008, 2007, 1007, 'Accepted');
INSERT INTO applications VALUES(3009, 2008, 1008, 'Pending');
INSERT INTO applications VALUES(3010, 2009, 1009, 'Rejected');
INSERT INTO applications VALUES(3011, 2010, 1010, 'Accepted');
INSERT INTO applications VALUES(3012, 2006, 1006, 'Rejected');

--Displaying table data using SELECT command
select * from companies where category = 'Multinational Technology';

--Displaying table data using subquery
select * from companies where company_id = (select company_id from jobs where job_id = 2006);

--Updating the data in a table
update jobs set job_title = 'Informatics Engineer' where job_id = 2008;

--Deleting row from a table
INSERT INTO jobs VALUES(2011, 'Data Scientist', 0010, 'Full Time', 25000, 'BSc in CSE');
delete from jobs where job_id = 2011;

--union, intersect, and except
select company_name from companies where company_name like 'S%' union select company_name from companies where company_name like '%o%';
select company_name from companies where company_name like 'S%' INTERSECT select
company_name from companies where company_name like '%o%';
select company_name from companies where company_name like 'S%' EXCEPT select
company_name from companies where company_name like '%o%';


--With clause
with max_salary(val) as (select max(salary) from jobs)
select * from jobs, max_salary where jobs.salary = max_salary.val;


--Aggregate function
--count how many row exist
select count(*) from companies;
--give alias name to any output in select command
select count(category) as number_of_categories from companies;
--count distinct
select count(distinct category) as number_of_categories from companies;
--count average and total
select avg(salary) from jobs;
select sum(salary) from jobs;
--find max and min
select max(salary) from jobs;
select min(salary) from jobs;


--Group by and Having
select job_type, avg(salary) from jobs group by job_type;
select job_type, avg(salary) from jobs group by job_type having avg(salary) > 18000;


--Nested subquery
 select company_name from companies where company_id =
 (select company_id from jobs where job_id =
 (select job_id from applications where app_id = 3003));


 --Set Membership (AND, OR, NOT)
select * from companies where category = 'Multinational Technology' and company_id in (select company_id from jobs where education like '%CSE%');
select * from jobs where salary > some(select salary from jobs where salary >= 20000);
select * from jobs where salary > all(select salary from jobs where salary >= 20000);
select * from jobs where salary >= 20000 and exists(select * from companies where category like '%Multinational Technology%');


--String operations
--beginning with H
SELECT * FROM jobs where education like 'H%';
--ending with e
SELECT * FROM jobs where education like '%E';
--contains c & E
SELECT * FROM jobs where education like '%C%E%';
--character length 3
SELECT * FROM jobs where education like '___';
--character length 3 or 10
SELECT * FROM jobs where education like '___' or education like '__________';


--Join operations
--natural JOIN
select * from companies natural join jobs where company_id = 0001;
select * from companies natural join jobs;
--join using
select company_name, job_title from companies join jobs using(company_id);
```

```
--on and relation
select company_name, job_title from companies join jobs on companies.company_id =
jobs.company_id;
--left outer join
select company_name, job_title from companies left outer join jobs using(company_id);
select company_name, job_title from companies left outer join jobs on
companies.company_id = jobs.company_id;
--right outer join
select company_name, job_title from companies right outer join jobs using(company_id);
--full outer join
select company_name, job_title from companies full outer join jobs using(company_id);


--Views
drop VIEW custom;
drop VIEW company_details;
drop VIEW Apple_Jobs;
--without other ATTRIBUTES
create view company_details as select company_id, company_name from companies;
SELECT * from company_details;
--as a combination
create view Apple_Jobs as select job_title from jobs where company_id = (select
company_id from companies where company_name = 'Apple');
SELECT * from Apple_Jobs;
--Using Other Views
create view custom as select * from company_details where company_id >= 0006;
SELECT * from custom;
```

PL/SQL:

```
--PL/SQL variable declaration and print value
set serveroutput on
declare
job_id jobs.job_id%type;
job_title jobs.job_title%type;
salary number;
begin
select job_id, job_title, salary into job_id, job_title, salary from jobs where job_id = 2007;
dbms_output.put_line('job_id: '||job_id|| ' job_title: '||job_title || ' salary: '||salary);
end;
/

--Insert and set default value
set serveroutput on
declare
app_id applications.app_id%type := 3013;
job_id applications.job_id%type := 2007;
user_id applications.user_id%type := 1008;
status applications.status%type := 'Pending';
begin
insert into applications values(app_id,job_id,user_id,status);
end;
/

--Row type
set serveroutput on
declare
```

```
job_row jobs%rowtype;
begin
select job_id, job_title, salary into job_row.job_id, job_row.job_title, job_row.salary from jobs where job_id = 2007;
dbms_output.put_line('job_id: '||job_row.job_id|| ' job_title: '||job_row.job_title || ' salary: '||job_row.salary);
end;
/


--Cursor and row count
set serveroutput on
declare
cursor job_cursor is select * from jobs;
job_row jobs%rowtype;
begin
open job_cursor;
fetch job_cursor into job_row.job_id, job_row.job_title, job_row.company_id, job_row.job_type, job_row.salary, job_row.education;
while job_cursor%found loop
dbms_output.put_line('job_id: '||job_row.job_id|| ' job_title: '||job_row.job_title || ' company_id: ' ||job_row.company_id|| ' job_type: ' ||job_row.job_type || ' salary: ' ||job_row.salary || ' education: ' ||job_row.education);
dbms_output.put_line('Row count: '|| job_cursor%rowcount);
fetch job_cursor into job_row.job_id,job_row.job_title,job_row.company_id,job_row.job_type,job_row.salary, job_row.education;
end loop;
close job_cursor;
end;
/
```

```
--FOR LOOP/WHILE LOOP/ARRAY with extend() function
set serveroutput on
declare
  counter number;
  job_title2 jobs.job_title%type;
  TYPE TITLEARRAY IS VARRAY(10) OF jobs.job_title%type;
  TITLE TITLEARRAY:=TITLEARRAY();
begin
  counter:=1;
  for x in 2001..2010
  loop
    select job_title into job_title2 from jobs where job_id=x;
    TITLE.EXTEND();
    TITLE(counter):=job_title2;
    counter:=counter+1;
  end loop;
  counter:=1;
  WHILE counter<=TITLE.COUNT
    LOOP
    DBMS_OUTPUT.PUT_LINE(TITLE(counter));
    counter:=counter+1;
  END LOOP;
end;
/

--ARRAY without extend() function
DECLARE
   counter NUMBER := 1;
   job_title2 jobs.job_title%TYPE;
   TYPE TITLEARRAY IS VARRAY(10) OF jobs.job_title%TYPE;
   TITLE TITLEARRAY:=TITLEARRAY('job 1', 'job 2', 'job 3', 'job 4', 'job 5', 'job 6', 'job 7', 'job 8', 'job 9', 'job 10');
BEGIN
   counter := 1;
   FOR x IN 2001..2010
   LOOP
     SELECT job_title INTO job_title2 FROM jobs WHERE job_id=x;
     TITLE(counter) := job_title2;
     counter := counter + 1;
   END LOOP;
   counter := 1;
   WHILE counter <= TITLE.COUNT
   LOOP
     DBMS_OUTPUT.PUT_LINE(TITLE(counter));
     counter := counter + 1;
   END LOOP;
END;

--IF /ELSEIF /ELSE
DECLARE
   counter NUMBER := 1;
   job_title2 jobs.job_title%TYPE;
   TYPE TITLEARRAY IS VARRAY(10) OF jobs.job_title%TYPE;
   TITLE TITLEARRAY:=TITLEARRAY('job 1', 'job 2', 'job 3', 'job 4', 'job 5', 'job 6', 'job 7', 'job 8', 'job 9', 'job 10');
BEGIN
   counter := 1;
   FOR x IN 2001..2010
   LOOP
```

```
      SELECT job_title INTO job_title2 FROM jobs WHERE job_id=x;
      if job_title2 = 'Software Developer'
        then
        dbms_output.put_line(job_title2||' is a '||'CSE or SWE Job');
      elsif job_title2='Chip Designer'
        then
        dbms_output.put_line(job_title2||' is a '||'EEE job');
      else
        dbms_output.put_line(job_title2||' is a '||'other job');
        end if;
   END LOOP;
END;

--Procedure
CREATE OR REPLACE PROCEDURE proc(
  var1 IN NUMBER,
  var2 OUT VARCHAR2,
  var3 IN OUT NUMBER
)
AS
  t_show CHAR(30);
BEGIN
  t_show := 'From procedure: ';
  SELECT job_title INTO var2 FROM jobs WHERE job_id IN (SELECT job_id FROM applications WHERE app_id = var1);
  var3 := var1 + 1;
  DBMS_OUTPUT.PUT_LINE(t_show || var2 || ' code is ' || var1 || ' In out parameter: ' || var3);
END;
/
```

```
--main
set serveroutput on
declare
app_id applications.app_id%type := 3001;
job_title jobs.job_title%type;
extra number;
begin
proc(app_id, job_title, extra);
end;
/

--Function
set serveroutput on
create or replace function fun(var1 in varchar) return varchar AS
value jobs.job_title%type;
begin
  select job_title into value from jobs where job_id = var1;
   return value;
end;
/
--main
set serveroutput on
declare
value varchar(20);
begin
value := fun(2001);
dbms_output.put_line('function returns ' || value);
end;
/
```

```
--drop procedure and function

drop procedure proc;

drop function fun;
```