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

CAREER GUIDANCE

19.05.2022

— SUBMITTED TO MS.PRAGYA MISHRA

102003611 NAVDEEP SINGH
102003016 ANIKET IDNANI
102003606 UJJAYANT KADIAN
102003354 KRISH MEHTA

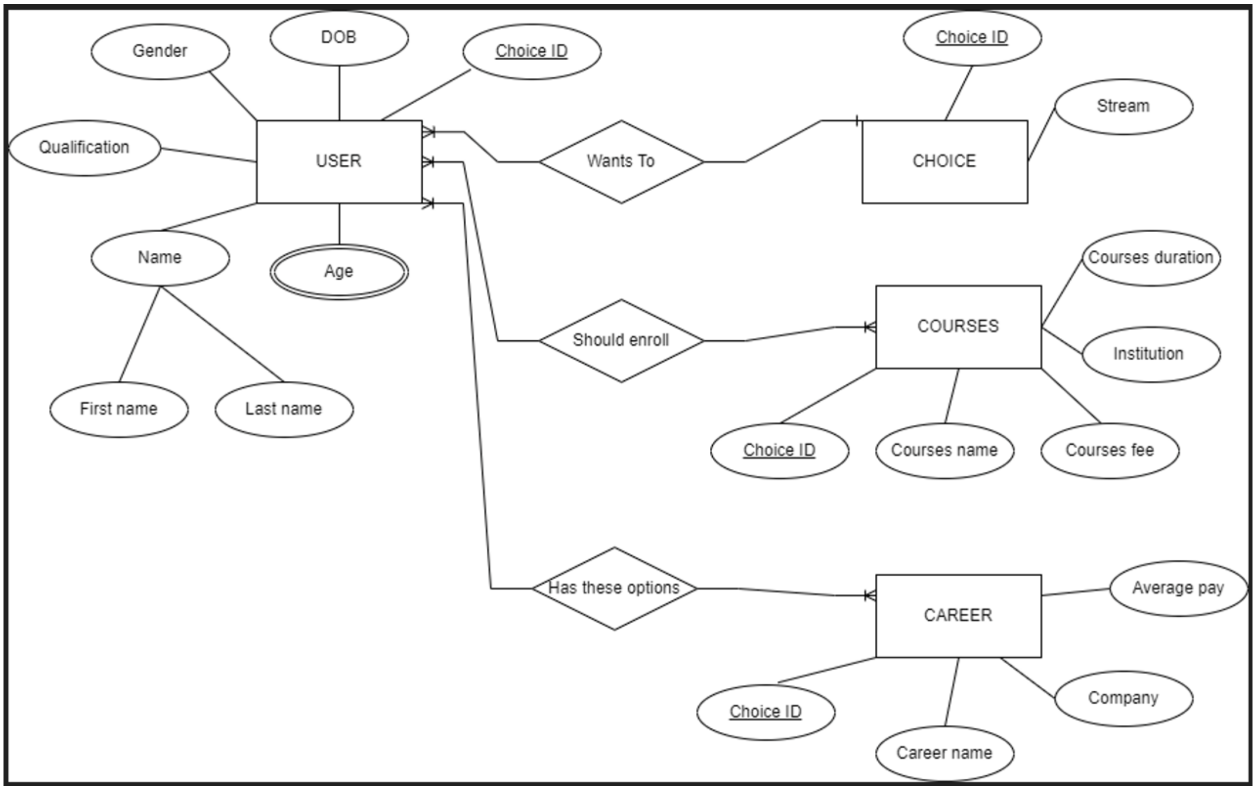
INDEX

PROBLEM STATEMENT

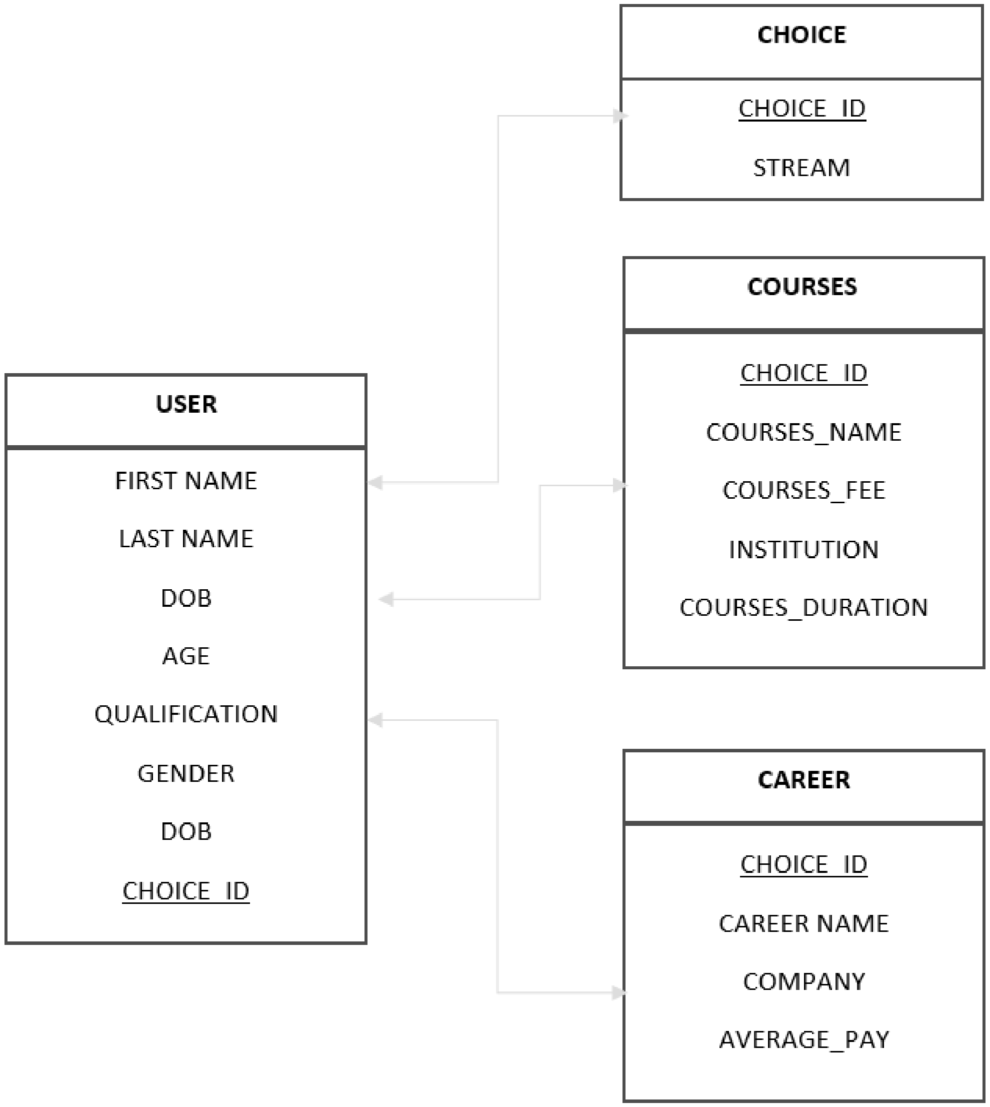
In order to comprehend the goal of demographic divide, each national of the country must be educated for employment, be it a person, a doctor, engineer or no matter to fulfill not solely the wants of our society however additionally the opposite societies elsewhere within the world. each student aspires to become a good person and self sustaining, however their passion and likings ar various. so as to cater to the aspirations of the scholars and facilitate them to aim high, they're needed to learn concerning the amount and sort of jobs this country has within the future years, further because the variety of business persons, entrepreneurs, and personnel altogether areas of employment. A national info of actual needs of personnel, category-wise by each Government and trade, the kind of instructional or ability qualification needed, nature of job, eligibility criteria, wage and perks, career development /progression in each field / sector of employment is desperately needed.

As every individual nurtures completely different needs to become in their life, there's a requirement for career dendrogram, in order that one will place forth a conjunctive efforts to become the simplest in his vocation and stand out in life instead of build many variety of mistakes of wrong choice in fact of education or career and repent later and prolong making an attempt various things losing his precious time/life.

ER DIAGRAM


Gender
DOB
Choice ID
Choice ID
Stream
Qualification
USER
Wants To
CHOICE
Name
Age
Courses duration
Should enroll
COURSES
Institution
First name
Last name
Choice ID
Courses name
Courses fee
Has these options
CAREER
Average pay
Choice ID
Company
Career name

ER DIAGRAM TO TABLE



CHOICE

CHOICE_ID

STREAM

COURSES

CHOICE_ID

COURSES_NAME

COURSES_FEE

INSTITUTION

COURSES_DURATION

USER

FIRST NAME

LAST NAME

DOB

AGE

QUALIFICATION

GENDER

DOB

CHOICE_ID

CAREER

CHOICE_ID

CAREER NAME

COMPANY

AVERAGE_PAY

NORMALIZED TABLES

USERS

USERS
USER_ID
FIRST_NAME
LAST_NAME
QUALIFICATION
GENDER
DOB
CHOICE_ID

USER_QUALIFICATION
USER_ID
QUALIFICATION
PRIMARY KEY

USER_PHONE
USER_ID
PHONE_NUMBER

CHOICE

CHOICE_STREAM
CHOICE_ID
STREAM

COURSES

COURSES
CORUSE_ID
COURSE_NAME

CHOICE_COURSE
CHOICE_ID
COURSE_ID

COURSE_INSTITUTION
COURSE_ID
INSTITUTION

COURSE_DETAILS
COURSE_ID
COURSE_DURATION
COURSE FEE

CAREER

CAREERS
CAREER_ID
CAREER_NAME

CHOICE_CAREER
CHOICE_ID
CAREER_ID

CAREER_DETAILS
CAREER_ID
CAREER_NAME
AVERAGE_PAY

PL/SQL CODES TO IMPLEMENT PROJECT

--stream table

```
CREATE TABLE `careergudiance`.`choice_stream` (
  `choice_id` INT NOT NULL,
  `stream` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`choice_id`),
  UNIQUE INDEX `stream_UNIQUE` (`stream` ASC) VISIBLE);
```

--users

```
CREATE TABLE `careergudiance`.`users` (
  `user_id` INT NOT NULL,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NULL,
  `qualification` VARCHAR(45) NULL,
  `gender` VARCHAR(45) NOT NULL,
  `dob` DATE NOT NULL,
  `choice_id` INT NOT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE INDEX `choice_id_UNIQUE` (`choice_id` ASC) VISIBLE,
  CONSTRAINT `choice_id`
    FOREIGN KEY (`choice_id`)
    REFERENCES `careergudiance`.`choice_stream` (`choice_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
ALTER TABLE `careergudiance`.`users`
DROP COLUMN `qualification`;
```

--user_qualification

```
CREATE TABLE `careergudiance`.`user_qualification` (
  `user_id` INT NOT NULL,
  `qualification` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`user_id`),
  CONSTRAINT `user_qualification`
    FOREIGN KEY (`user_id`)
    REFERENCES `careergudiance`.`users` (`user_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
```

--user_phone number

```
CREATE TABLE `careergudiance`.`user_phone` (
  `user_id` INT NOT NULL,
  `phone_number` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`user_id`, `phone_number`),
  CONSTRAINT `user_phone`
    FOREIGN KEY (`user_id`)
    REFERENCES `careergudiance`.`users` (`user_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
ALTER TABLE `careergudiance`.`user_phone`
CHANGE COLUMN `phone_number` `phone_number` INT NOT NULL ;
```

--courses

```
CREATE TABLE `careergudiance`.`courses` (
  `course_id` INT NOT NULL,
  `course_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`course_id`));
```

--choice_courses

```
CREATE TABLE `careergudiance`.`choice_course` (
  `choice_id` INT NOT NULL,
  `course_id` INT NOT NULL,
  PRIMARY KEY (`course_id`, `choice_id`),
  INDEX `choice_courses_idx` (`choice_id` ASC) VISIBLE,
  CONSTRAINT `choice_courses`
    FOREIGN KEY (`choice_id`)
    REFERENCES `careergudiance`.`courses` (`course_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
ALTER TABLE `careergudiance`.`choice_course`
DROP FOREIGN KEY `choice_courses`;
ALTER TABLE `careergudiance`.`choice_course`
ADD INDEX `choice_courses_idx` (`choice_id` ASC) VISIBLE,
DROP INDEX `choice_courses_idx` ;
;
ALTER TABLE `careergudiance`.`choice_course`
ADD CONSTRAINT `choice_courses`
  FOREIGN KEY (`choice_id`)
  REFERENCES `careergudiance`.`choice_stream` (`choice_id`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
ADD CONSTRAINT `courses_name`
  FOREIGN KEY (`course_id`)
  REFERENCES `careergudiance`.`courses` (`course_id`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
```

--course institution

```
CREATE TABLE `careergudiance`.`course_institution` (
  `course_id` INT NOT NULL,
  `institution` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`course_id`),
  CONSTRAINT `course_institution`
    FOREIGN KEY (`course_id`)
    REFERENCES `careergudiance`.`courses` (`course_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
```

--course details

```
CREATE TABLE `careergudiance`.`course_details` (
  `course_id` INT NOT NULL,
  `course_duration(in hours)` INT NOT NULL,
  `course_fee` INT NULL,
  PRIMARY KEY (`course_id`),
 CONSTRAINT `course_coursedetails`
  FOREIGN KEY (`course_id`)
  REFERENCES `careergudiance`.`courses` (`course_id`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT);
```

--careers

```
CREATE TABLE `careergudiance`.`careers` (
 `career_id` INT NOT NULL,
 `career_name` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`career_id`));
```

--choice_career

```
CREATE TABLE `careergudiance`.`choice_career` (
 `choice_id` INT NOT NULL,
 `career_id` INT NOT NULL,
 PRIMARY KEY (`choice_id`, `career_id`),
 INDEX `career_careername_idx` (`career_id` ASC) VISIBLE,
 CONSTRAINT `choice_choicestream`
  FOREIGN KEY (`choice_id`)
  REFERENCES `careergudiance`.`choice_stream` (`choice_id`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
 CONSTRAINT `career_careername`
  FOREIGN KEY (`career_id`)
  REFERENCES `careergudiance`.`careers` (`career_id`)
```

```
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
```

--career_details

```
CREATE TABLE `careergudiance`.`career_details` (
  `career_id` INT NOT NULL,
  `career_name` VARCHAR(45) NOT NULL,
  `average_pay` INT NOT NULL,
  PRIMARY KEY (`career_id`, `career_name`),
  CONSTRAINT `careerdetails_careername`
    FOREIGN KEY (`career_id`)
    REFERENCES `careergudiance`.`careers` (`career_id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT);
ALTER TABLE `careergudiance`.`user_phone` 
CHANGE COLUMN `phone_number` `phone_number` BIGINT(10) NOT NULL ;
ALTER TABLE `careergudiance`.`user_qualification` 
DROP PRIMARY KEY,
ADD PRIMARY KEY (`user_id`, `qualification`);
;
ALTER TABLE `careergudiance`.`career_details` 
CHANGE COLUMN `career_name` `company` VARCHAR(45) NOT NULL ;
ALTER TABLE `careergudiance`.`course_details` 
CHANGE COLUMN `course_duration(in hours)` `course_duration` INT NOT NULL ;
```

CODE TO CHECK PROJECT FUNCTIONALITY

```
set @id := 2;


-- Stream Chosen:
SELECT a.stream
FROM choice_stream as a
INNER JOIN users as b ON a.choice_id=b.choice_id
WHERE b.user_id = @id;


-- Courses that you have to take:
SELECT c.course_name
FROM users as a
INNER JOIN choice_course as b ON a.choice_id=b.choice_id
INNER JOIN courses as c ON b.course_id=c.course_id
WHERE a.user_id = @id;

-- Course details:
SELECT e.course_name, c.course_duration, c.course_fee, d.institution
FROM users as a
INNER JOIN choice_course as b ON a.choice_id=b.choice_id
INNER JOIN course_details as c ON b.course_id=c.course_id
INNER JOIN course_institution as d ON b.course_id = d.course_id
INNER JOIN courses as e ON b.course_id = e.course_id
WHERE a.user_id = @id;

-- Career Options:
SELECT e.career_name
FROM users as a
INNER JOIN choice_career as d ON a.choice_id = d.choice_id
INNER JOIN careers as e ON d.career_id = e.career_id
```

```
WHERE a.user_id = @id;
```

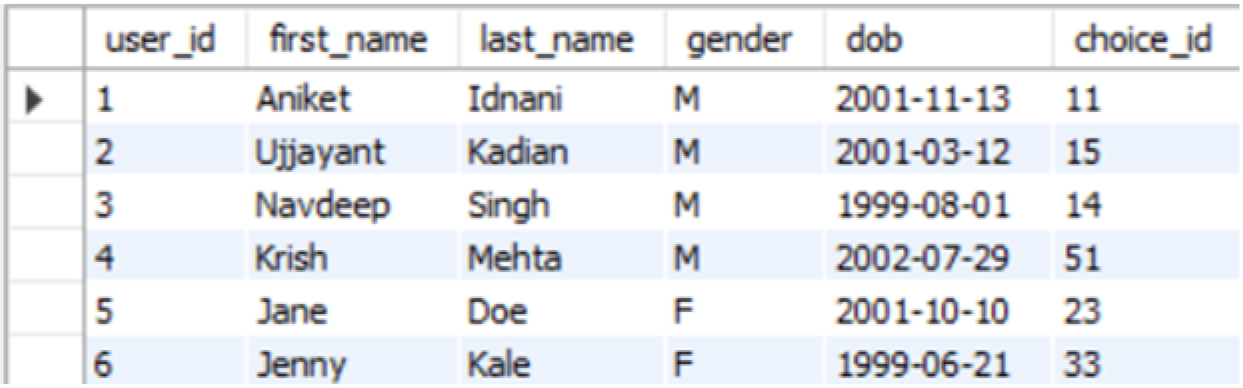
-- Career Details

```
SELECT c.career_name, d.company, d.average_pay
FROM users as a
INNER JOIN choice_career as b ON a.choice_id = b.choice_id
INNER JOIN careers as c ON b.career_id = c.career_id
INNER JOIN career_details as d ON b.career_id = d.career_id
WHERE a.user_id = @id
```

OUTPUT SCREENSHOTS

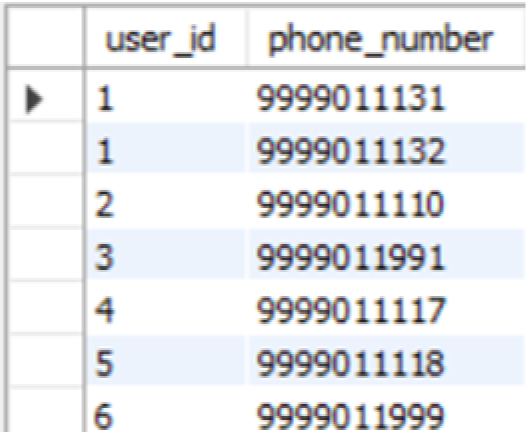
TABLES CREATED IN SQL

Users:



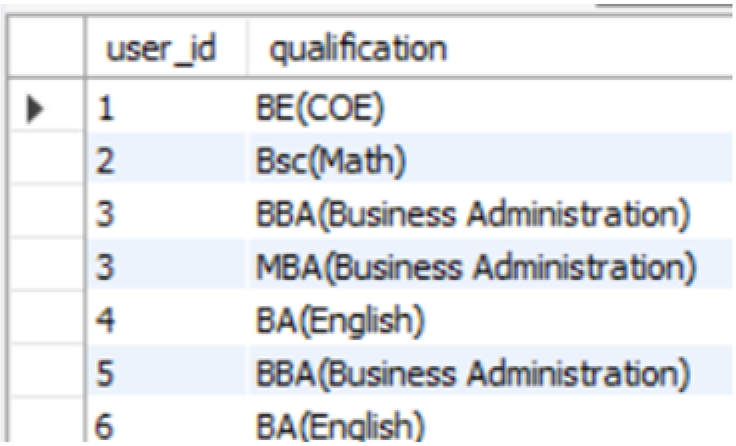
	user_id	first_name	last_name	gender	dob	choice_id
▶	1	Aniket	Idnani	M	2001-11-13	11
	2	Ujjayant	Kadian	M	2001-03-12	15
	3	Navdeep	Singh	M	1999-08-01	14
	4	Krish	Mehta	M	2002-07-29	51
	5	Jane	Doe	F	2001-10-10	23
	6	Jenny	Kale	F	1999-06-21	33

User_phone:



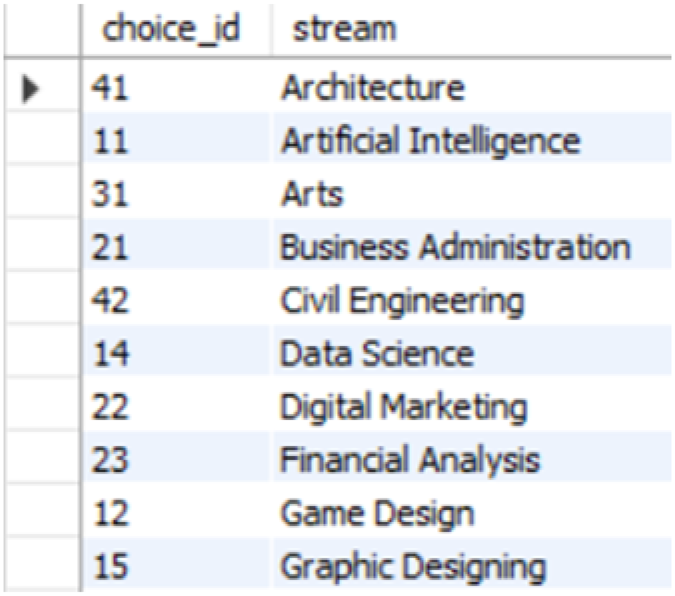
	user_id	phone_number
▶	1	9999011131
	1	9999011132
	2	9999011110
	3	9999011991
	4	9999011117
	5	9999011118
	6	9999011999

User_qualification:



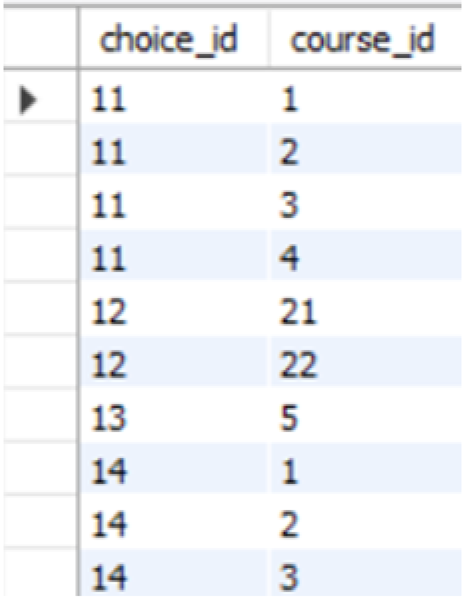
	user_id	qualification
▶	1	BE(COE)
	2	Bsc(Math)
	3	BBA(Business Administration)
	3	MBA(Business Administration)
	4	BA(English)
	5	BBA(Business Administration)
	6	BA(English)

Choice_stream:



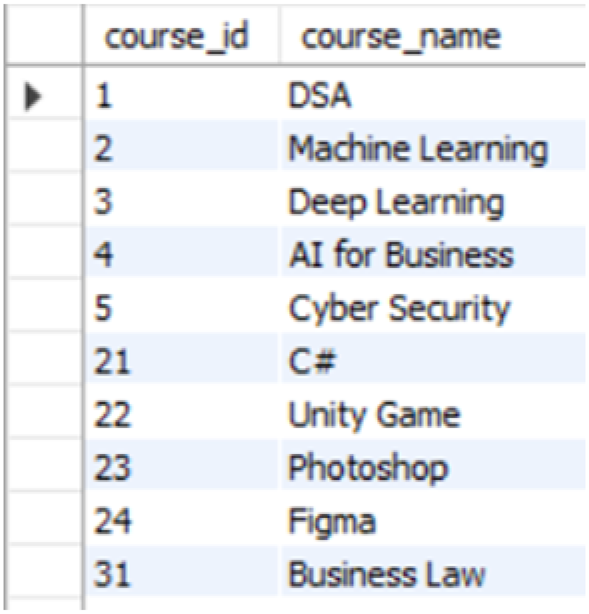
choice_id	stream
41	Architecture
11	Artificial Intelligence
31	Arts
21	Business Administration
42	Civil Engineering
14	Data Science
22	Digital Marketing
23	Financial Analysis
12	Game Design
15	Graphic Designing

Choice_course



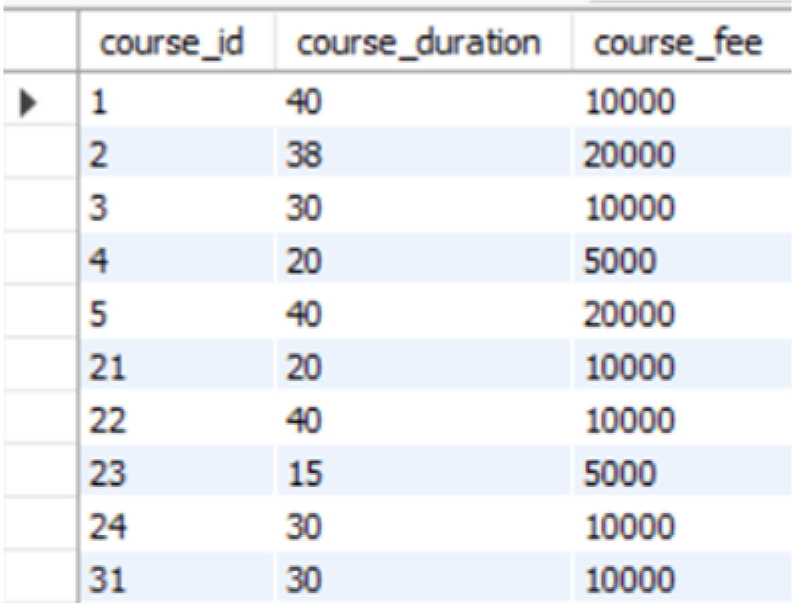
choice_id	course_id
11	1
11	2
11	3
11	4
12	21
12	22
13	5
14	1
14	2
14	3

Courses:



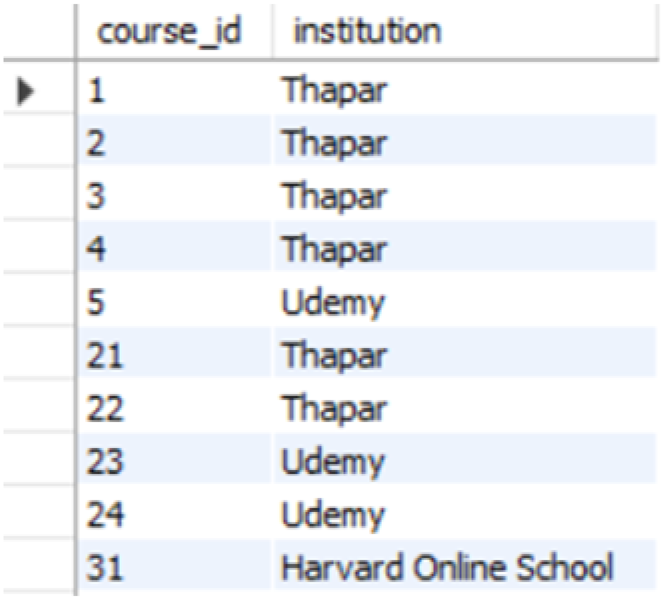
	course_id	course_name
▶	1	DSA
	2	Machine Learning
	3	Deep Learning
	4	AI for Business
	5	Cyber Security
	21	C#
	22	Unity Game
	23	Photoshop
	24	Figma
	31	Business Law

Course_details:



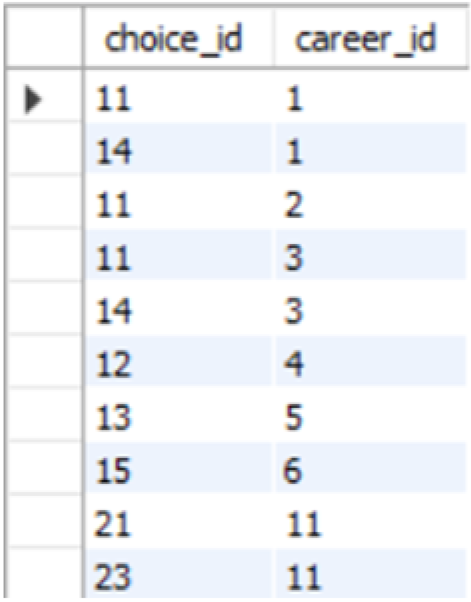
	course_id	course_duration	course_fee
▶	1	40	10000
	2	38	20000
	3	30	10000
	4	20	5000
	5	40	20000
	21	20	10000
	22	40	10000
	23	15	5000
	24	30	10000
	31	30	10000

Course_institution:



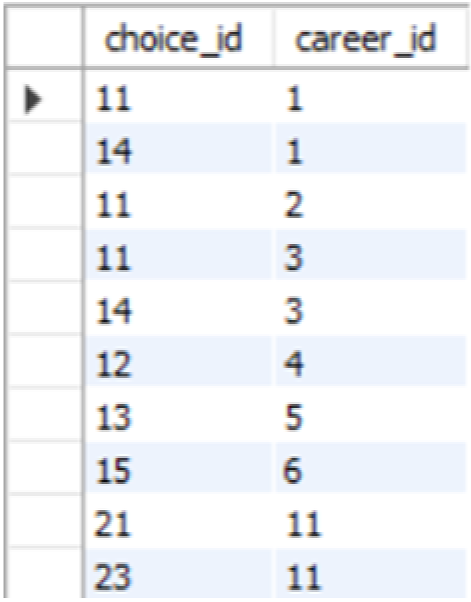
course_id	institution
1	Thapar
2	Thapar
3	Thapar
4	Thapar
5	Udemy
21	Thapar
22	Thapar
23	Udemy
24	Udemy
31	Harvard Online School

Choice_career:



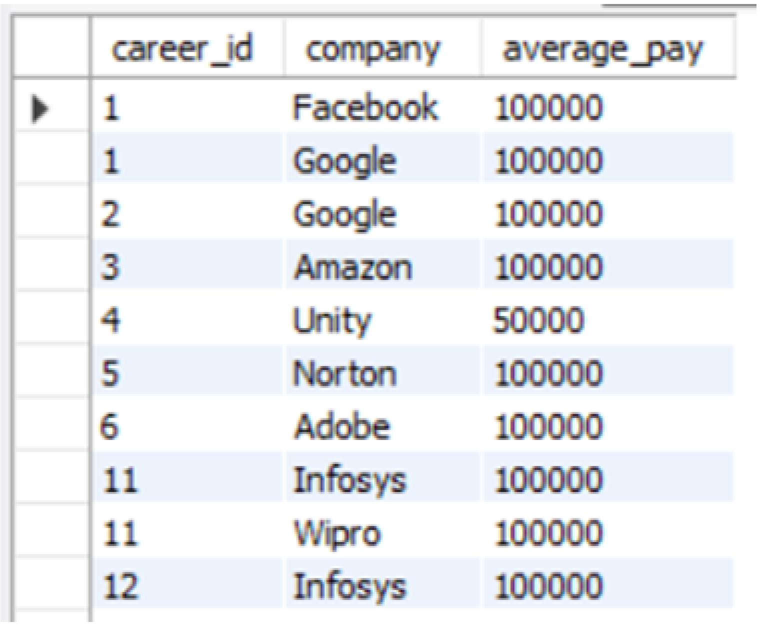
choice_id	career_id
11	1
14	1
11	2
11	3
14	3
12	4
13	5
15	6
21	11
23	11

Careers:



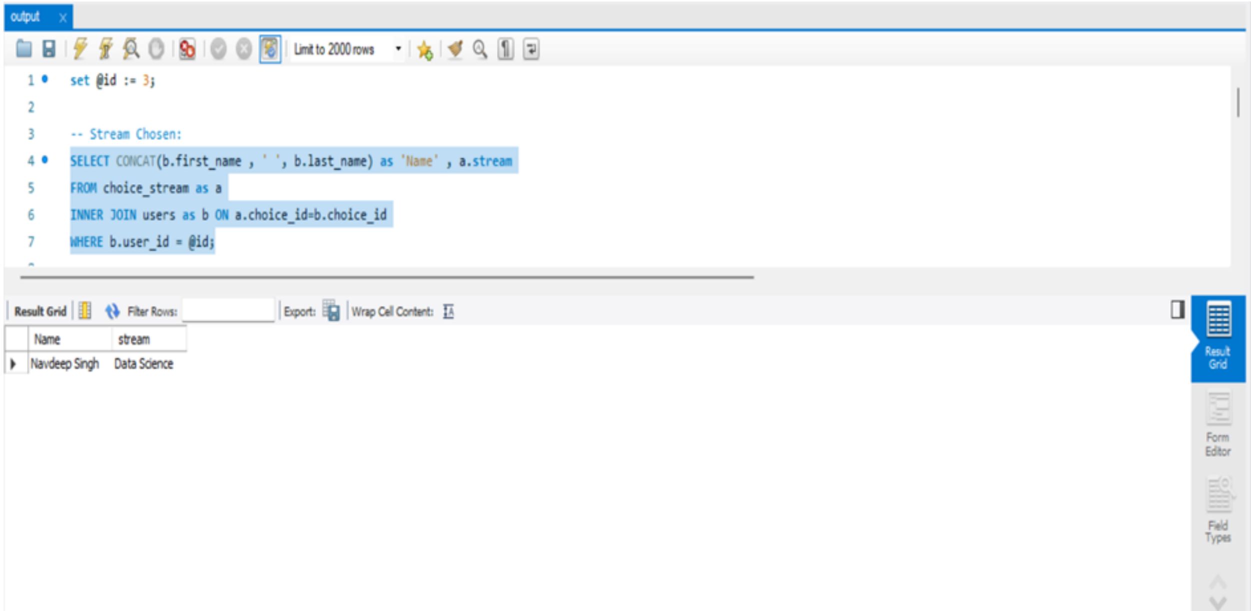
	choice_id	career_id
▶	11	1
	14	1
	11	2
	11	3
	14	3
	12	4
	13	5
	15	6
	21	11
	23	11

Career details:



	career_id	company	average_pay
▶	1	Facebook	100000
	1	Google	100000
	2	Google	100000
	3	Amazon	100000
	4	Unity	50000
	5	Norton	100000
	6	Adobe	100000
	11	Infosys	100000
	11	Wipro	100000
	12	Infosys	100000

Output results:

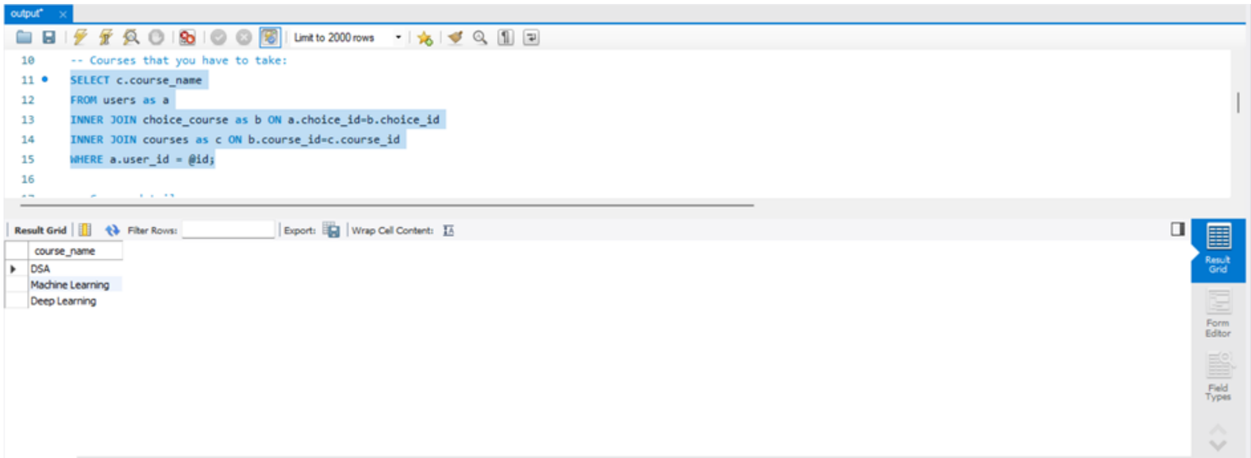


```
set @id := 3;

-- Stream Chosen:
SELECT CONCAT(b.first_name , ' ', b.last_name) as 'Name' , a.stream
FROM choice_stream as a
INNER JOIN users as b ON a.choice_id=b.choice_id
WHERE b.user_id = @id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

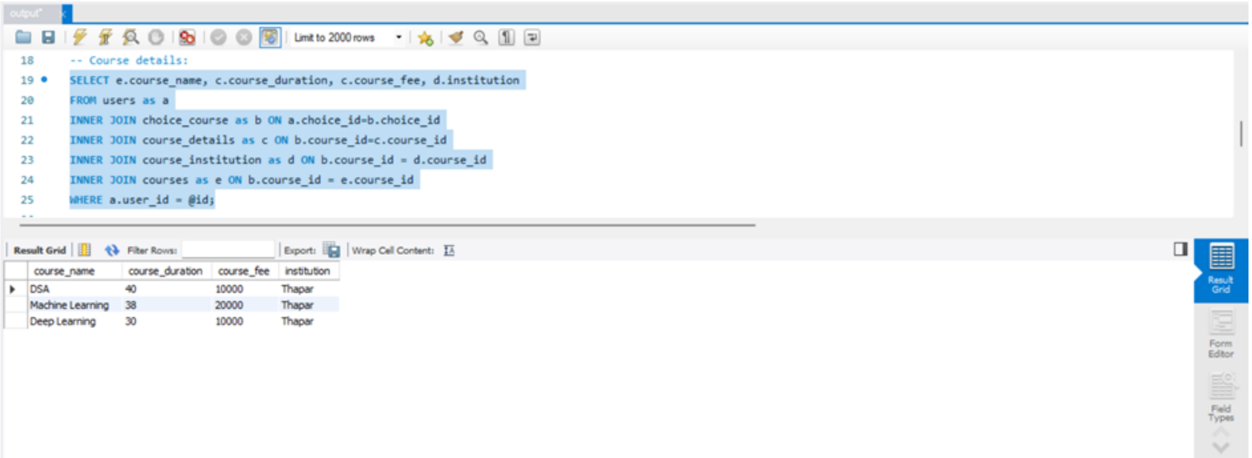
Name	stream
Navdeep Singh	Data Science



```
-- Courses that you have to take:
SELECT c.course_name
FROM users as a
INNER JOIN choice_course as b ON a.choice_id=b.choice_id
INNER JOIN courses as c ON b.course_id=c.course_id
WHERE a.user_id = @id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

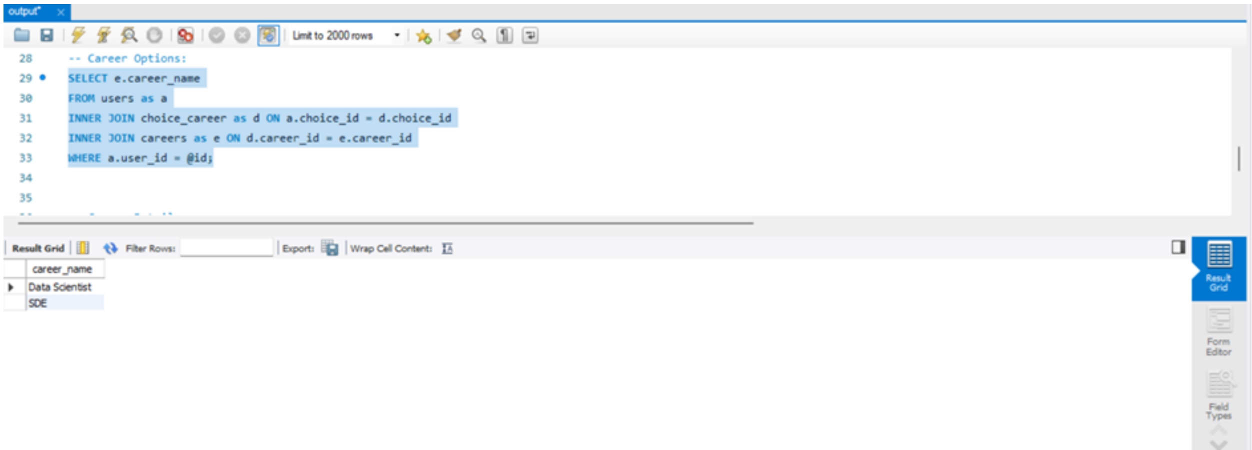
course_name
DSA
Machine Learning
Deep Learning



```
-- Course details:
SELECT e.course_name, c.course_duration, c.course_fee, d.institution
FROM users as a
INNER JOIN choice_course as b ON a.choice_id=b.choice_id
INNER JOIN course_details as c ON b.course_id=c.course_id
INNER JOIN course_institution as d ON b.course_id = d.course_id
INNER JOIN courses as e ON b.course_id = e.course_id
WHERE a.user_id = @id;
```

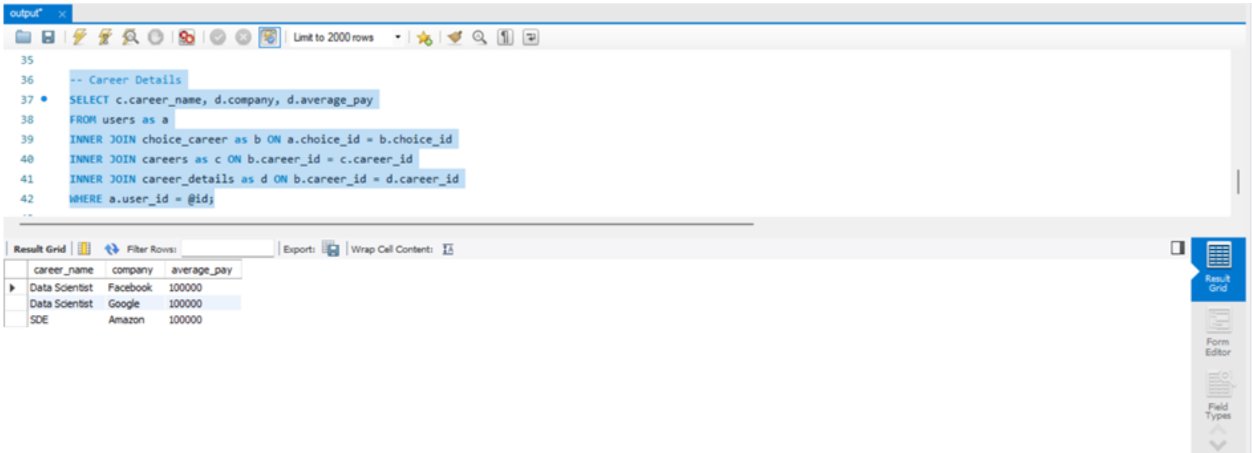
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_name	course_duration	course_fee	institution
DSA	40	10000	Thapar
Machine Learning	38	20000	Thapar
Deep Learning	30	10000	Thapar



```
-- Career Options:
SELECT e.career_name
FROM users as a
INNER JOIN choice_career as d ON a.choice_id = d.choice_id
INNER JOIN careers as e ON d.career_id = e.career_id
WHERE a.user_id = @id;
```

career_name
Data Scientist
SDE



```
-- Career Details
SELECT c.career_name, d.company, d.average_pay
FROM users as a
INNER JOIN choice_career as b ON a.choice_id = b.choice_id
INNER JOIN careers as c ON b.career_id = c.career_id
INNER JOIN career_details as d ON b.career_id = d.career_id
WHERE a.user_id = @id;
```

career_name	company	average_pay
Data Scientist	Facebook	100000
Data Scientist	Google	100000
SDE	Amazon	100000