Introduction to Databases

Assignment 2

Nicholas Hopewell 0496633

1.

I will first show how to create the tables with MySQL:

Create “Conference” table:

```
CREATE TABLE Conference(
    ConferenceID INT NOT NULL,
    Name VARCHAR(60),
    Date DATE,
    Time TIME,
    Location VARCHAR(100),

    PRIMARY KEY(ConferenceID)
)
```

Create “Presentations” table:

```
CREATE TABLE Presentations(
    PresentationID INT NOT NULL,
    subject_area VARCHAR(60),
    paper_title VARCHAR(100),
    time TIME,
    location VARCHAR(100),

    PRIMARY KEY(PresentationID)
);
```

Creating table “Academic” which contains fkeys:

```sql
CREATE TABLE Academic(
    MemberID INT NOT NULL AUTO_INCREMENT,
    PresentationID INT NOT NULL,
    ConferenceID INT NOT NULL,
    first_name VARCHAR(100),
    last_name VARCHAR(250),
    paper_title VARCHAR(100),

    PRIMARY KEY(MemberID),
    INDEX(PresentationID),

    FOREIGN KEY(PresentationID)
        REFERENCES Presentations(PresentationID)
         ON UPDATE CASCADE ON DELETE RESTRICT,

    FOREIGN KEY (ConferenceID)
        REFERENCES Conference(ConferenceID)
        ON UPDATE CASCADE ON DELETE RESTRICT
);
```

HTML rendered:

Presenter Information:

First Name: [ ]

Last Name: [ ]

Member ID: [ ]

Conference Information (check all to confirm):

Confirm Conference: [x] Annual Database Conference 2018

Confirm Date: [x] 2018-08-05

Confirm Time: [x] 9:00

Confirm Location: [x] 255 Front St W, Toronto, ON

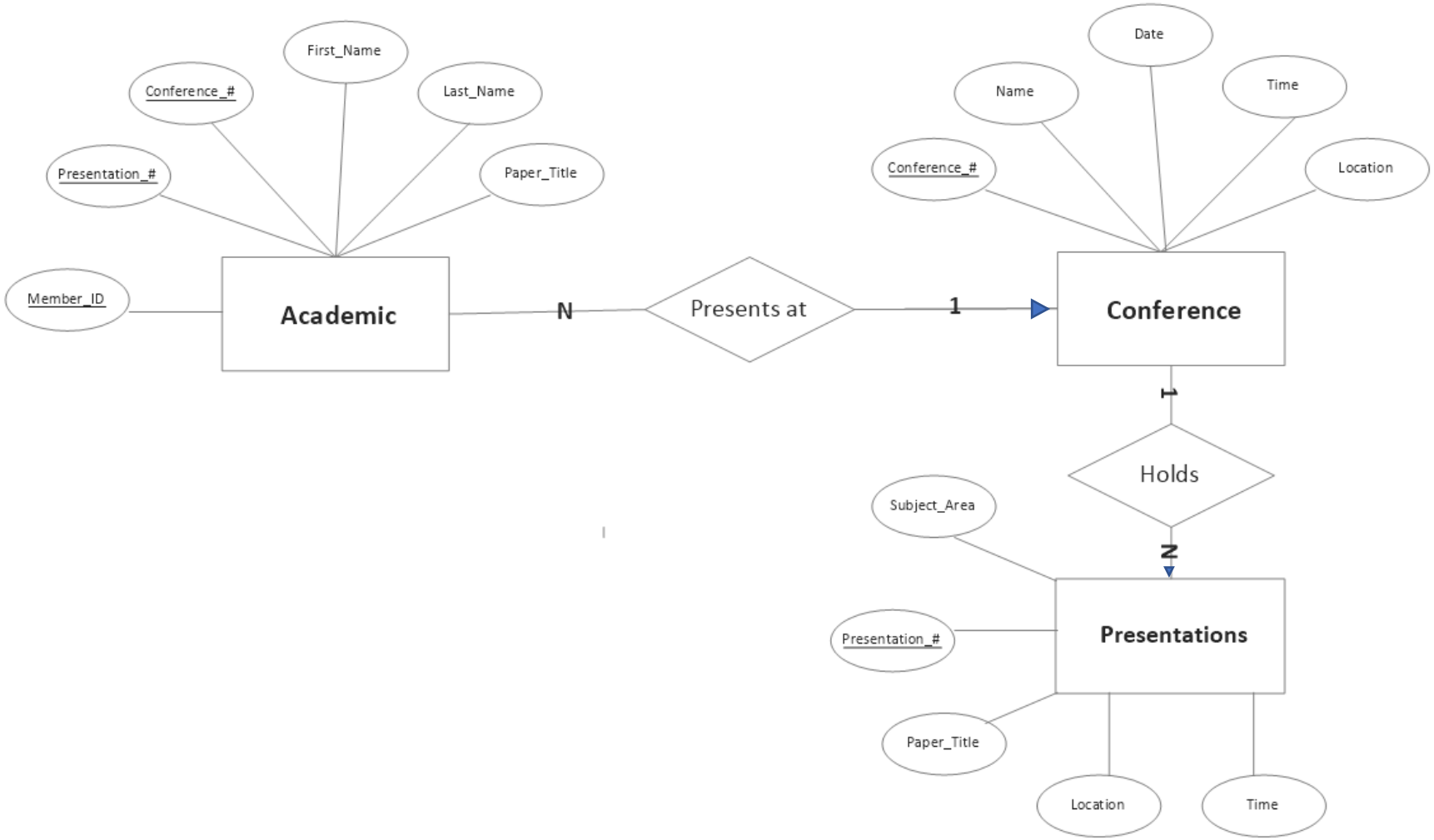
Presentation Information (check all to confirm):

Subject area: [Computer science ▾]

Presentation #: [ ]

Paper Title: [ ]

Submit Information: [Submit]


First_Name
Conference_#
Last_Name
Presentation_#
Paper_Title
Member_ID
Academic
N
Presents at
1
Date
Name
Time
Conference_#
Location
Conference
1
Holds
N
Subject_Area
Presentations
Presentation_#
Paper_Title
Location
Time

2.

** I am using PostgreSQL for this question since it was not specified, and I would really like to practice PSQL. *First*, I will create the three tables. I will create the User table last as it contains the fkeys used to link the 3 tables.

<u>Create Occupation table:</u>

db_assignment_2 on postgres@PostgreSQL 10

```sql
CREATE TABLE public.Occupation(
    OccupationId SERIAL NOT NULL PRIMARY KEY,
    OccupationName CHARACTER VARYING(100)
);
```

Insert into Occupation table:

db_assignment_2 on postgres@PostgreSQL 10

```sql
INSERT INTO public.Occupation(occupationname)
VALUES ('Software Engineer'),
       ('Accountant'),
       ('Pharmacist'),
       ('Library Assistant');
```

<u>Note:</u> Because I specify id as type SERIAL I get an auto incrementing int starting from 1 (don't need to manually insert id)

Verify table:

db_assignment_2 on postgres@PostgreSQL 10

```sql
SELECT * FROM public.Occupation;
```

Data Output | Explain | Messages | Notifications | Query History

	occupationid integer	occupationname character varying (100)
1	1	Software Engineer
2	2	Accountant
3	3	Pharmacist
4	4	Library Assistant

Create City table:

db_assignment_2 on postgres@PostgreSQL 10

```
CREATE TABLE City(
    CityId SERIAL NOT NULL PRIMARY KEY,
    CityName CHARACTER VARYING(40)
);
```

Insert into City table:

db_assignment_2 on postgres@PostgreSQL 10

```
INSERT INTO public.City(cityname)
VALUES ('Halifax'),
       ('Calgary'),
       ('Boston'),
       ('New York'),
       ('Toronto');
```

Verify table:

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT * FROM public.City;
```

Data Output | Explain | Messages | Notifications | Query History

	cityid integer	cityname character varying (40)
1	1	Halifax
2	2	Calgary
3	3	Boston
4	4	New York
5	5	Toronto

Create User table:

db_assignment_2 on postgres@PostgreSQL 10

```sql
CREATE TABLE public.User(
    id SERIAL NOT NULL,
    name CHARACTER VARYING(100),
    AGE INTEGER,
    GENDER CHARACTER VARYING(6),
    OccupationId INTEGER NOT NULL,
    CityId INTEGER NOT NULL,
    CONSTRAINT user_pkey PRIMARY KEY (id),
    CONSTRAINT fk_occupation_id FOREIGN KEY (OccupationId)
         REFERENCES public.Occupation (OccupationId) MATCH SIMPLE
         ON UPDATE NO ACTION
         ON DELETE NO ACTION,
    CONSTRAINT fk_city_id FOREIGN KEY (CityId)
         REFERENCES public.City (CityId) MATCH SIMPLE
         ON UPDATE NO ACTION
         ON DELETE NO ACTION
);
```

Insert into User table:

db_assignment_2 on postgres@PostgreSQL 10

```sql
INSERT INTO public.User(name, age, gender, occupationid, cityid)
VALUES ('John', 25, 'Male', 1, 3),
       ('Sara', 20, 'Female', 3, 4),
       ('Victor', 31, 'Male', 2, 5),
       ('Jane', 27, 'Female', 1, 3);
```

Verify table:

db_assignment_2 on postgres@PostgreSQL 10

```sql
SELECT * FROM public.User;
```

Data Output | Explain | Messages | Notifications | Query History

	id integer	name character varying (100)	age integer	gender character varying (6)	occupationid integer	cityid integer
1	1	John	25	Male	1	3
2	2	Sara	20	Female	3	4
3	3	Victor	31	Male	2	5
4	4	Jane	27	Female	1	3

2.1)

a)

	name character varying (100)
1	Victor
2	Jane

b)

Data Output | Explain | Messages | Notifications | Query History

	id integer	name character varying (100)	age integer	gender character varying (6)	occupationid integer	cityid integer
1	1	John	25	Male	1	3
2	2	Sara	20	Female	3	4
3	3	Victor	31	Male	2	5
4	4	Jane	27	Female	1	3

c)

Data Output | Explain | Messages | Notifications | Query History

	id integer	name character varying (100)	age integer	gender character varying (6)	occupationid integer	cityid integer	occupationid integer	occupationname character varying (100)
1	1	John	25	Male	1	3	1	Software Engineer
2	2	Sara	20	Female	3	4	3	Pharmacist
3	3	Victor	31	Male	2	5	2	Accountant
4	4	Jane	27	Female	1	3	1	Software Engineer

d)

Data Output | Explain | Messages | Notifications | Query History

	cityid integer	occupationid integer	id integer	name character varying (100)	age integer	gender character varying (6)	occupationname character varying (100)	cityname character varying (40)
1	3	1	1	John	25	Male	Software Engineer	Boston
2	4	3	2	Sara	20	Female	Pharmacist	New York
3	5	2	3	Victor	31	Male	Accountant	Toronto
4	3	1	4	Jane	27	Female	Software Engineer	Boston

e)

Data Output | Explain | Messages | Notifications | Quer

	name character varying (100)	gender character varying (6)
1	Jane	Female
2	John	Male

2.2)

*<u>Note:</u> I will be referencing the **default public schema and the table** within each query even though it can be left out in many cases. This is just for practice.*

a)

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT name
FROM public.User
WHERE age > 25;
```

b)

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT *
FROM public.User
WHERE id > 2 OR Age != 31;
```

c)

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT *
FROM public.User, public.Occupation
WHERE public.User.OccupationID = public.Occupation.OccupationId;
```

d)

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT *
FROM public.User NATURAL JOIN public.occupation NATURAL JOIN public.City;
```

- Here I am using a natural join option on the same named keys which I linked earlier

e)

db_assignment_2 on postgres@PostgreSQL 10

```
SELECT public.User.name, public.User.gender
FROM public.User INNER JOIN public.City ON public.City.cityid = public.User.cityid
WHERE public.City.cityname = 'Boston';
```

- Here I am showing above that one can do this sort of thing with an INNER JOIN on fkeys instead of a natural join to make it clearer what is happening. Natural join would give the same result.