CPSC 304 Project Cover Page

Milestone #: 3

Date: Sep 26th, 2024

Group Number: 29

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Ziqing Wang	22270649	g0h7c	g0h7c@ugrad.cs.ubc.ca
Owen Zheng	35933183	y6i3e	zcc2280411284@gmail.com
Jiawei Hu	57536633	i5m2m	i5m2m@ugrad.cs.ubc.ca

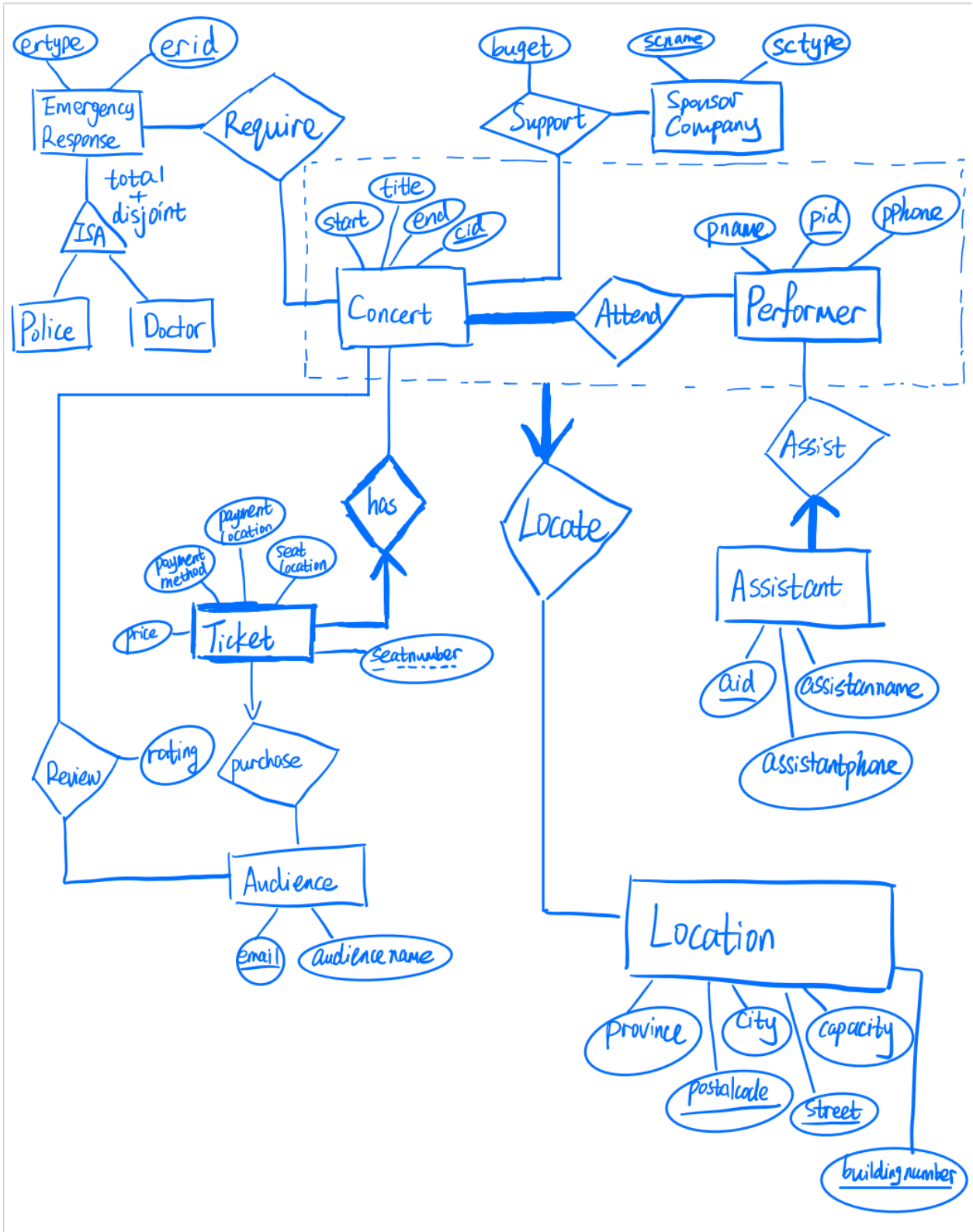
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary

Our project features ticketing services of various concert venues. Users are able to book tickets, view the concert information and rate concerts according to our database specifications.

ER Diagram


ertype
erid
Emergency Response
Require
total + disjoint
ISA
Police
Doctor
buget
sname
sctype
Support
Sponsor Company
title
start
end
cid
Concert
Attend
pname
pid
phone
Performer
Assist
Assistant
aid
assistantname
assistantphone
has
Locate
payment location
payment method
seat location
price
Ticket
seatnumber
Review
rating
purchase
Audience
email
audiencename
Location
province
city
capacity
postalcode
street
buildingnumber

Changes from Milestone 1 for ER Diagram:

Reason to change: To be more descriptive. For the last point, just to add more FD to work with.

- **Performers** “attend” **Concert** rather than “has” a concert.
- erid and ertype for **Emergency Response** rather than type and id.
- cid for **Concert** instead of id.
- audiencename for **Audience** instead of name.
- pid, pname, pphone for **Performer** instead id, name & phone.
- assistant name, assistantphone for **Assistant** instead of name & phone.
- scname, sctype instead of name, type for **SponsorCompany**.
- seatnumber instead of seat#.
- buildingnumber instead of building#.
- Added seatlocation for **Ticket**, province and city for **Location** because we want to add more FD to the relational database.

Reason to not change:

- We keep the PK of the Location unchanged. The example on page 8 of the slide inspired us. We think only postalcode, street and building# can uniquely identify the Location. We also hope that the points deducted because of this could be earned back from milestone 1 because this was a working example illustrated in slides.

RELATIONAL MODEL

EmergencyResponse (erid: NUMBER(4,0), ertype: VARCHAR2(50))

PK (erid)

CK (erid)

Police (erid: NUMBER(4,0))

PK (erid)

CK (erid)

FK (erid) (NOTE: reference to EmergencyResponse)

Doctor (erid: NUMBER(4,0))

PK (erid)

CK (erid)

FK (erid) (NOTE: reference to EmergencyResponse)

Require (erid: NUMBER(4,0), cid: NUMBER(4,0))

PK (erid, cid)

CK (erid, cid)

FK (erid) (NOTE: reference to EmergencyResponse)

FK (cid) (NOTE: reference to Concert)

Audience (email: VARCHAR2(50), audiencename: VARCHAR2(50))

PK (email)

CK (email)

Review (email: VARCHAR2(50), cid: NUMBER(4,0), rating: NUMBER(1,0))

PK (email, cid)

CK (email, cid)

FK (email) (NOTE: reference to Audience)

FK (cid) (NOTE: reference to Concert)

Ticket_Purchase_Has (seatnumber: NUMBER(4,0), price: NUMBER(6,2), paymentmethod: VARCHAR2(50), paymentlocation: VARCHAR2(50), email: VARCHAR2(50), cid: NUMBER(4,0), seatlocation: VARCHAR2(50))

PK (seatnumber, cid)

CK (seatnumber, cid)

FK (email) (NOTE: reference to Audience)

FK (cid) (NOTE: reference to Concert)

SponsorCompany (scname: VARCHAR2(50), sctype: VARCHAR2(50))

PK (scname)

CK (scname)

Support (scname: VARCHAR2(50), cid: NUMBER(4,0), budget: NUMBER(10,2))

PK (scname, cid)

CK (scname, cid)

FK (scname) (NOTE: reference to SponsorCompany)

FK (cid) (NOTE: reference to Concert)

Concert (cid: NUMBER(4,0), start: DATE, end: DATE, title: VARCHAR2(50))

PK (cid)

CK (cid)

Total participation can not be modelled for now

Performer (pid: NUMBER(4,0), pname: VARCHAR2(50), pphone: CHAR(10))

PK (pid)

CK (pid)

Attend_Locate (cid: NUMBER(4,0), pid: NUMBER(4,0), postalcode: CHAR(7), street: VARCHAR2(50), buildingnumber: NUMBER(5,0))

PK (cid, pid)

CK (cid, pid)

FK (cid) (NOTE: reference to Concert)

FK (pid) (NOTE: reference to Performer)

FK (postalcode, street, buildinnumbergnumber) (NOTE: reference to Location)

postalcode NOT NULL

street NOT NULL

buildingnumbernumber NOT NULL

Location (postalcode: CHAR(7), street: VARCHAR2(50), buildingnumber: NUMBER(5,0), capacity: NUMBER(6,0), city: VARCHAR2(50), province: VARCHAR2(50))

PK (postalcode, street, buildingnumber)

CK (postalcode, street, buildingnumber)

Assistant_Assist (aid: NUMBER(4,0), assistantname: VARCHAR2(50), assistantphone: CHAR (10), pid: NUMBER(4,0))

PK (aid)

CK (aid)

FK (pid) (NOTE: reference to Performer)

pid NOT NULL

FDs

EmergencyResponse: erid **->** ertype

Police: No FDs

Doctor: No FDs

Require: No FDs

Audience: email -> audiencename

Review: email, cid -> rating

Ticket_Purchase_Has:

Seatnumber, cid -> paymentmethod

Seatnumber, cid -> paymentlocation

Seatnumber, cid -> email

Seatnumber, cid -> price

Seatnumber, cid -> seatlocation

seatlocation -> price

SponsorCompany: scname -> sctype

Support: scname, cid -> budget

Concert:

cid -> start

cid -> end

cid -> title

Performer:

pid -> pname

pid -> pphone

Attend_Locate:

cid, pid -> postalcode

cid, pid -> street

cid, pid -> building#

Location:

postalcode, street, buildingnumber -> capacity

postalcode, street, buildingnumber -> city

postalcode, street, buildingnumber -> province

postalcode -> city

postalcode -> province

street, city -> province

Assistant_Assist:

aid -> assistantname

aid -> assistantphone

aid -> pid

We intended to turn all tables into 3NF. (Steps shown on the next page)

Table_Purchase_Has: Lossless join method

Location: Synthesis Method

Ticket-Purchase-Has

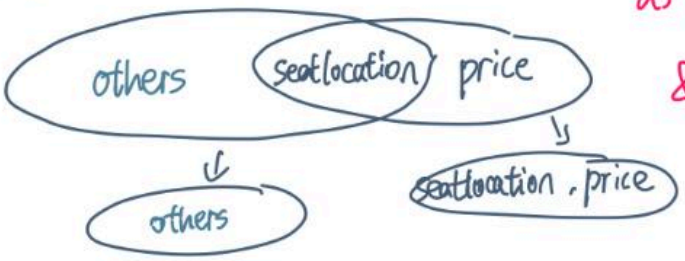
1. in the FD, seatlocation → price, seatlocation isn't a superkey & price isn't a part of the key.
   ∴ It's not in 3NF

2. ① Done. All RHS contain only 1 attr
   ② Done. Because LHS already minimized
   ③ Delete seatnumber, cid → price b/c price can be implicitly determined by seatnumber, cid.

3. Decompose from reduced FDs.

seatnumber, cid$^+$ = {seatnumber, cid, paymentmethod, paymentlocation, email, price, seatlocation}

seatlocation$^+$ = {seatlocation, price} violates 3DNF as price isn't part of key & seatlocation isn't a super key



Take all key's combination's closure that is in FD.

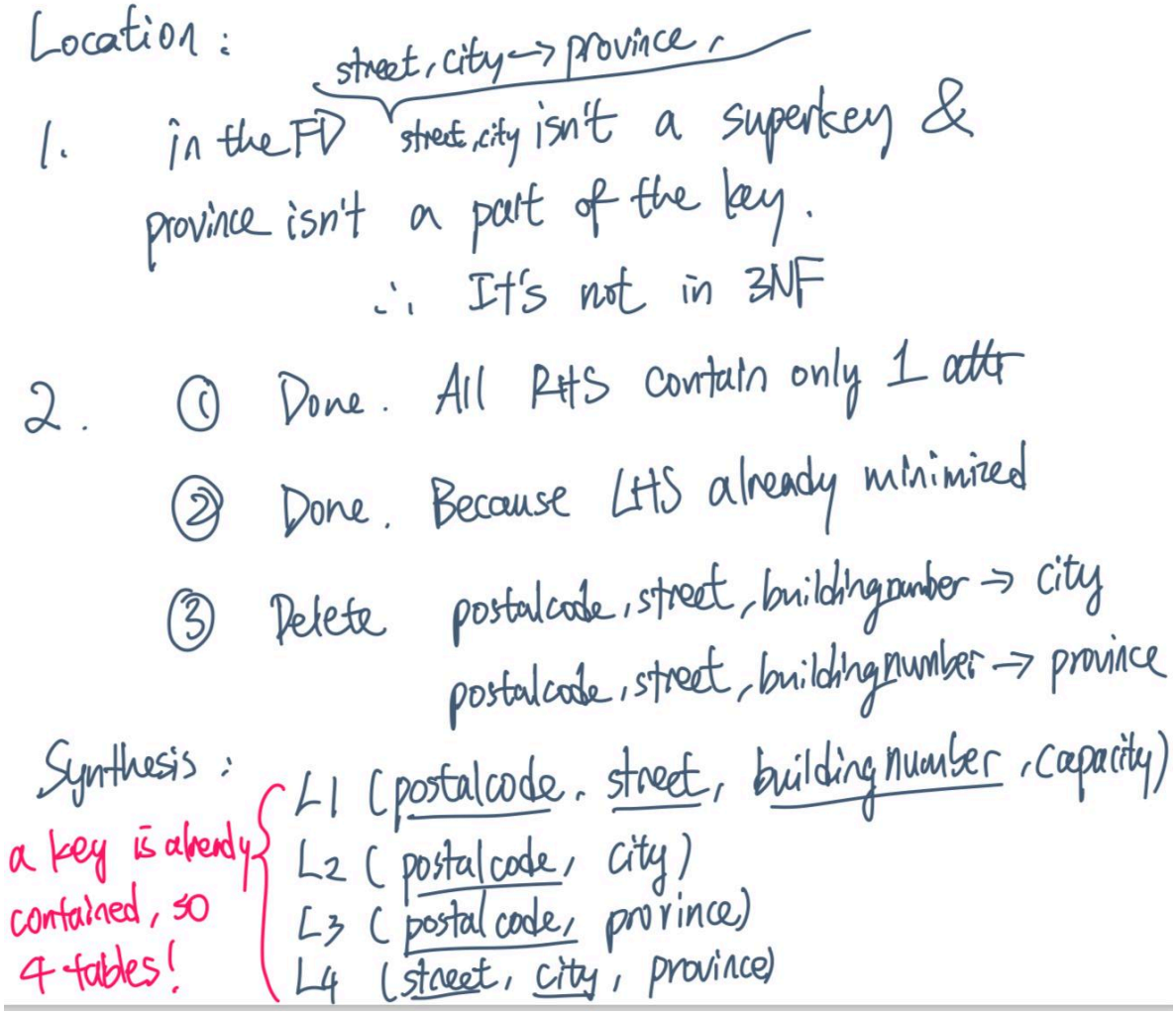
seatnumber$^+$ = {seatnumber} } trivial; and no more other interesting closure to take
cid$^+$ = {cid}

∴ TPH1

payment method, payment location, email, seatlocation
r seatnumber, cid

TPH2

seatlocation, price



Resulting 3NF Tables

1. EmergencyResponse (erid: NUMBER(4, 0), ertype: VARCHAR2(50))

PK (erid)

CK (erid)

2. Police (erid: NUMBER(4, 0))

PK (erid)

CK (erid)

FK (erid) (NOTE: reference to EmergencyResponse)

3. Doctor (erid: NUMBER(4, 0))

PK (erid)

CK (erid)

FK (erid) (NOTE: reference to EmergencyResponse)

4. Require (erid: NUMBER(4, 0), cid: NUMBER(4, 0))

PK (erid, cid)

CK (erid, cid)

FK (erid) (NOTE: reference to EmergencyResponse)

FK (cid) (NOTE: reference to Concert)

5. Audience (email: VARCHAR2(50), audiencename: VARCHAR2(50))

PK (email)

CK (email)

6. Review (email: VARCHAR2(50), cid: NUMBER(4, 0), rating: NUMBER(1, 0))

PK (email, cid)

CK (email, cid)

FK (email) (NOTE: reference to Audience)

FK (cid) (NOTE: reference to Concert)

7. TPH1 (seatnumber: NUMBER(4, 0), cid: NUMBER(4, 0), paymentmethod: VARCHAR2(50), paymentlocation: VARCHAR2(50), email: VARCHAR2(50), seatlocation: VARCHAR2(50))

PK (seatnumber, cid)

CK (seatnumber, cid)

FK (seatlocation) (NOTE: reference to TPH2)

FK (cid) (NOTE: reference to Concert)

FK (email) (NOTE: reference to Audience)

8. TPH2 (seatlocation: VARCHAR2(50), price: NUMBER(6, 2))

PK (seatlocation)

CK (seatlocation)

9. L1 (postalcode: CHAR(7), street: VARCHAR2(50), buildingnumber: NUMBER(5, 0), capacity: NUMBER(6, 0))

PK (postalcode, street, buildingnumber)

CK (postalcode, street, buildingnumber)

FK (postalcode) (NOTE: reference to L2)

10. L2 (postalcode: CHAR(7), city: VARCHAR2(50))

PK (postalcode)

CK (postalcode)

FK (postalcode) (NOTE: reference to L3)

11. L3 (postalcode: CHAR(7), province: VARCHAR2(50))

PK (postalcode)

CK (postalcode)

12. L4 (street: VARCHAR2(50), city: VARCHAR2(50), province: VARCHAR2(50))

PK (street, city)

CK (street, city)

13. SponsorCompany (scname: VARCHAR2(50), sctype: VARCHAR2(50))

PK (scname)

CK (scname)

14. Support (scname: VARCHAR2(50), cid: NUMBER(4,0), budget: NUMBER(10, 2))

PK (scname, cid)

CK (scname, cid)

FK (scname) (NOTE: reference to SponsorCompany)

FK (cid) (NOTE: reference to Concert)

15. Concert (cid: NUMBER(4, 0), start: DATE, end: DATE, title: VARCHAR2(50))

PK (cid)

CK (cid)

Total participation can not be modeled for now

16. Performer (pid: NUMBER(4, 0), pname: VARCHAR2(50), pphone: CHAR(10))

PK (pid)

CK (pid)

17. Attend_Locate (cid: NUMBER(4, 0), pid: NUMBER(4, 0), postalcode: CHAR(7), street: VARCHAR2(50), buildingnumber: NUMBER(5, 0))

PK (cid, pid)

CK (cid, pid)

FK (cid) (NOTE: reference to Concert)

FK (pid) (NOTE: reference to Performer)

FK (postalcode, street, buildingnumber) (NOTE: reference to L1)

postalcode NOT NULL

street NOT NULL

buildingnumber NOT NULL

18. Assistant_Assist (aid: NUMBER(4, 0), assistantname: VARCHAR2(50), assistantphone: CHAR(10), pid: NUMBER(4, 0))

PK (aid)

CK (aid)

FK (pid) (NOTE: reference to Performer)

pid NOT NULL

SQL Table Creation

```
-- Create the EmergencyResponse table
CREATE TABLE EmergencyResponse (
  erid NUMBER(4, 0),
  ertype VARCHAR2(50),
  PRIMARY KEY (erid)
);
```

```sql
-- Create the Police table
CREATE TABLE Police (
  erid NUMBER(4, 0),
  PRIMARY KEY (erid),
  FOREIGN KEY (erid) REFERENCES EmergencyResponse(erid)
);

-- Create the Doctor table
CREATE TABLE Doctor (
  erid NUMBER(4, 0),
  PRIMARY KEY (erid),
  FOREIGN KEY (erid) REFERENCES EmergencyResponse(erid)
);

-- Create the Concert table
CREATE TABLE Concert (
  cid NUMBER(4, 0),
  starttime DATE,
  endtime DATE,
  title VARCHAR2(50),
  PRIMARY KEY (cid)
);
```

```sql
-- Create the Audience table
CREATE TABLE Audience (
  email VARCHAR2(50),
  audiencename VARCHAR2(50),
  PRIMARY KEY (email)
);

-- Create the Require table
CREATE TABLE Require (
  erid NUMBER(4, 0),
  cid NUMBER(4, 0),
  PRIMARY KEY (erid, cid),
  FOREIGN KEY (erid) REFERENCES EmergencyResponse(erid),
  FOREIGN KEY (cid) REFERENCES Concert(cid)
);

-- Create the Review table
CREATE TABLE Review (
  email VARCHAR2(50),
  cid NUMBER(4, 0),
  rating NUMBER(1, 0),
  PRIMARY KEY (email, cid),
```

```
  FOREIGN KEY (email) REFERENCES Audience(email),
  FOREIGN KEY (cid) REFERENCES Concert(cid)
);

-- Create the TPH2 table
CREATE TABLE TPH2 (
  seatlocation VARCHAR2(50),
  price NUMBER(6, 2),
  PRIMARY KEY(seatlocation)
);

-- Create the TPH1 table
CREATE TABLE TPH1 (
  seatnumber NUMBER(4, 0),
  cid NUMBER(4, 0),
  paymentmethod VARCHAR2(50),
  paymentlocation VARCHAR2(50),
  email VARCHAR2(50),
  seatlocation VARCHAR2(50),
  PRIMARY KEY(seatnumber, cid),
  FOREIGN KEY (cid) REFERENCES Concert (cid),
  FOREIGN KEY (seatlocation) REFERENCES TPH2 (seatlocation),
  FOREIGN KEY (email) REFERENCES Audience(email)
```

```
);


-- Create L3 table
CREATE TABLE L3 (
  postalcode CHAR(7),
  province VARCHAR2(50),
  PRIMARY KEY(postalcode)
);


-- Create L2 table
CREATE TABLE L2 (
  postalcode CHAR(7),
  city VARCHAR2(50),
  PRIMARY KEY(postalcode),
  FOREIGN KEY(postalcode) REFERENCES L3(postalcode)
);


-- Create L1 table
CREATE TABLE L1 (
  postalcode CHAR(7),
  street VARCHAR2(50),
  buildingnumber NUMBER(5, 0), --changed!!!
  capacity NUMBER(6, 0),       --changed!!!
```

```
  PRIMARY KEY(postalcode, street, buildingnumber),
  FOREIGN KEY(postalcode) REFERENCES L2(postalcode)
);

-- Create L4 table
CREATE TABLE L4 (
  street VARCHAR2(50),
  city VARCHAR2(50),
  province VARCHAR2(50),
  PRIMARY KEY(street, city)
);

-- Create SponsorCompany table
CREATE TABLE SponsorCompany (
  scname VARCHAR2(50),
  sctype VARCHAR2(50),
  PRIMARY KEY (scname)
);

-- Create Support table
CREATE TABLE Support (
  scname VARCHAR2(50),
  cid NUMBER(4, 0),
```

```
  budget NUMBER(10, 2),
  PRIMARY KEY (scname, cid),
  FOREIGN KEY (scname) REFERENCES SponsorCompany(scname),
  FOREIGN KEY (cid) REFERENCES Concert(cid)
);


-- Create Performer table
CREATE TABLE Performer (
  pid NUMBER(4, 0),
  pname VARCHAR2(50),
  pphone CHAR(10),
  PRIMARY KEY (pid)
);


-- Create Attend_Locate table
CREATE TABLE Attend_Locate (
  cid NUMBER(4, 0),
  pid NUMBER(4, 0),
  postalcode CHAR(7) NOT NULL,
  street VARCHAR2(50) NOT NULL,
  buildingnumber NUMBER(5, 0) NOT NULL,
  PRIMARY KEY (cid, pid),
  FOREIGN KEY (cid) REFERENCES Concert(cid),
```

```
  FOREIGN KEY (pid) REFERENCES Performer(pid),

  FOREIGN KEY (postalcode, street, buildingnumber) REFERENCES
L1(postalcode, street, buildingnumber)

);



-- Create AssistantAssist table

CREATE TABLE AssistantAssist (

  aid NUMBER(4, 0),

  assistantname VARCHAR2(50),

  assistantphone CHAR(10),

  pid NUMBER(4, 0) NOT NULL,

  FOREIGN KEY (pid) REFERENCES Performer(pid)

);
```

SQL Table Insertion

```
-- Insert values into EmergencyResponse table
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1000, 'Security');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1001, 'Patrol');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1002, 'SWAT');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1003, 'Traffic
Police');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1004, 'K-9 Unit');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1005, 'Paramedic');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1006, 'First Aid');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1007, 'General
Practitioner');
```

```
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1008, 'Emergency
Physician');
INSERT INTO EmergencyResponse (erid, ertype) VALUES (1009,
'Pediatrician');

-- Insert values into Police table
INSERT INTO Police (erid) VALUES (1000);
INSERT INTO Police (erid) VALUES (1001);
INSERT INTO Police (erid) VALUES (1002);
INSERT INTO Police (erid) VALUES (1003);
INSERT INTO Police (erid) VALUES (1004);

-- Insert values into Doctor table
INSERT INTO Doctor (erid) VALUES (1005);
INSERT INTO Doctor (erid) VALUES (1006);
INSERT INTO Doctor (erid) VALUES (1007);
INSERT INTO Doctor (erid) VALUES (1008);
INSERT INTO Doctor (erid) VALUES (1009);

-- Insert values into Concert table
INSERT INTO Concert (cid, starttime, endtime, title) VALUES
(0001, TO_DATE('2024-11-01 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-11-01 21:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Rock Night
Festival');
INSERT INTO Concert (cid, starttime, endtime, title) VALUES
(0002, TO_DATE('2024-11-05 19:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-11-05 22:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Beethoven Late
Sonatas');
INSERT INTO Concert (cid, starttime, endtime, title) VALUES
(0003, TO_DATE('2024-11-10 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-11-10 20:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Jazz in the
Park');
INSERT INTO Concert (cid, starttime, endtime, title) VALUES
(0004, TO_DATE('2024-11-15 20:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-11-15 23:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Hip Pop
Evening');
INSERT INTO Concert (cid, starttime, endtime, title) VALUES
(0005, TO_DATE('2024-11-20 18:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-11-20 21:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Country Music
Night');
```

```sql
-- Insert values into Audience table
INSERT INTO Audience (email, audiencename) VALUES
('zcc2280411284@gmail.com', 'Chengchao Zheng');
INSERT INTO Audience (email, audiencename) VALUES
('winifred.wang2004@gmail.com', 'Ziqing Wang');
INSERT INTO Audience (email, audiencename) VALUES
('owen04@student.ubc.ca', 'Owen Zheng');
INSERT INTO Audience (email, audiencename) VALUES ('john04@gmail.com',
'John Smith');
INSERT INTO Audience (email, audiencename) VALUES ('2280411284@qq.com',
'Moumou Zheng');

-- Insert values into Require table
INSERT INTO Require (erid, cid) VALUES (1000, 0001);
INSERT INTO Require (erid, cid) VALUES (1001, 0001);
INSERT INTO Require (erid, cid) VALUES (1006, 0001);
INSERT INTO Require (erid, cid) VALUES (1006, 0002);
INSERT INTO Require (erid, cid) VALUES (1003, 0003);
INSERT INTO Require (erid, cid) VALUES (1006, 0003);
INSERT INTO Require (erid, cid) VALUES (1007, 0003);
INSERT INTO Require (erid, cid) VALUES (1006, 0005);

-- Insert values into Review table
INSERT INTO Review (email, cid, rating) VALUES ('zcc2280411284@gmail.com',
0001, 5);
INSERT INTO Review (email, cid, rating) VALUES
('winifred.wang2004@gmail.com', 0002, 2);
INSERT INTO Review (email, cid, rating) VALUES ('owen04@student.ubc.ca',
0001, 4);
INSERT INTO Review (email, cid, rating) VALUES ('john04@gmail.com', 0004,
1);
INSERT INTO Review (email, cid, rating) VALUES ('2280411284@qq.com', 0005,
3);

-- Insert values into TPH2 table
INSERT INTO TPH2(seatlocation, price) VALUES ('Runways', 1000.00);
INSERT INTO TPH2(seatlocation, price) VALUES ('Suite', 800.00);
INSERT INTO TPH2(seatlocation, price) VALUES ('First Floor', 500.00);
INSERT INTO TPH2(seatlocation, price) VALUES ('Second Floor', 300.00);
```

```
INSERT INTO TPH2(seatlocation, price) VALUES ('Third Floor', 100.00);

-- Insert values into TPH1 table
INSERT INTO TPH1(cid, seatnumber, seatlocation, email, paymentmethod,
paymentlocation) VALUES (0001, 0010, 'Runways', 'zcc2280411284@gmail.com',
'Visa', 'Offline');
INSERT INTO TPH1(cid, seatnumber, seatlocation, email, paymentmethod,
paymentlocation) VALUES (0002, 0011, 'Runways',
'winifred.wang2004@gmail.com', 'Debit', 'Online');
INSERT INTO TPH1(cid, seatnumber, seatlocation, email, paymentmethod,
paymentlocation) VALUES (0003, 0050, 'Suite', 'owen04@student.ubc.ca',
'Paypal', 'Online');
INSERT INTO TPH1(cid, seatnumber, seatlocation, email, paymentmethod,
paymentlocation) VALUES (0004, 0400, 'Second Floor', 'john04@gmail.com',
'Cash', 'Offline');
INSERT INTO TPH1(cid, seatnumber, seatlocation, email, paymentmethod,
paymentlocation) VALUES (0005, 0600, 'Third Floor', '2280411284@qq.com',
'Cash', 'Offline');

-- Insert values into L3 table
INSERT INTO L3(postalcode, province) VALUES ('V6T 1Z3', 'BC');
INSERT INTO L3(postalcode, province) VALUES ('V5K 0A3', 'BC');
INSERT INTO L3(postalcode, province) VALUES ('T5J 0H6', 'AB');
INSERT INTO L3(postalcode, province) VALUES ('H3B 5EB', 'QC');
INSERT INTO L3(postalcode, province) VALUES ('M5J 2X2', 'ON');

-- Insert values into L2 table
INSERT INTO L2(postalcode, city) VALUES ('V6T 1Z3', 'Vancouver');
INSERT INTO L2(postalcode, city) VALUES ('V5K 0A3', 'Vancouver');
INSERT INTO L2(postalcode, city) VALUES ('T5J 0H6', 'Edmonton');
INSERT INTO L2(postalcode, city) VALUES ('H3B 5EB', 'Montreal');
INSERT INTO L2(postalcode, city) VALUES ('M5J 2X2', 'Toronto');

-- Insert values into L1 table
INSERT INTO L1(postalcode, street, buildingnumber, capacity) VALUES ('V6T
1Z3', 'Thunderbird Blvd', 6066, 5054);
INSERT INTO L1(postalcode, street, buildingnumber, capacity) VALUES ('V5K
0A3', 'Griffiths Wy', 800, 19700);
INSERT INTO L1(postalcode, street, buildingnumber, capacity) VALUES ('T5J
0H6', '104 Ave NW', 10220, 20734);
```

```sql
INSERT INTO L1(postalcode, street, buildingnumber, capacity) VALUES ('H3B 5EB', 'Av. des Canadiens-de-Montreal', 1909, 21000);
INSERT INTO L1(postalcode, street, buildingnumber, capacity) VALUES ('M5J 2X2', 'Bay St', 40, 19800);

-- Insert values into L4 table
INSERT INTO L4(street, city, province) VALUES ('Thunderbird Blvd', 'Vancouver', 'BC');
INSERT INTO L4(street, city, province) VALUES ('Griffiths Wy', 'Vancouver', 'BC');
INSERT INTO L4(street, city, province) VALUES ('104 Ave NW', 'Edmonton', 'AB');
INSERT INTO L4(street, city, province) VALUES ('Av. des Canadiens-de-Montreal', 'Montreal', 'QC');
INSERT INTO L4(street, city, province) VALUES ('Bay st', 'Toronto', 'ON');

-- Insert values into SponsorCompany table
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Bank of Montreal', 'Finance');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Canadian Imperial Bank of Commerce', 'Finance');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Royal Bank of Canada', 'Finance');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Toronto-Dominion Bank', 'Finance');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Apple Inc.', 'Technology');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Microsoft Corporation', 'Technology');
INSERT INTO SponsorCompany (scname, sctype) VALUES ('Tencent', 'Entertainment');

-- Insert values into Support table
INSERT INTO Support (scname, cid, budget) VALUES ('Bank of Montreal', 0001, 100000.00);
INSERT INTO Support (scname, cid, budget) VALUES ('Royal Bank of Canada', 0002, 200000.00);
INSERT INTO Support (scname, cid, budget) VALUES ('Bank of Montreal', 0003, 300000.00);
```

```
INSERT INTO Support (scname, cid, budget) VALUES ('Apple Inc.', 0004,
100000.00);
INSERT INTO Support (scname, cid, budget) VALUES ('Tencent', 0005,
500000.00);

-- Insert values into Performer table
INSERT INTO Performer (pid, pname, pphone) VALUES (0001, 'Dave Grohl',
'5847526272');
INSERT INTO Performer (pid, pname, pphone) VALUES (0002, 'Anna Netrebko',
'2362516808');
INSERT INTO Performer (pid, pname, pphone) VALUES (0003, 'Gregory Porter',
'2046383438');
INSERT INTO Performer (pid, pname, pphone) VALUES (0004, 'Drake',
'7429333662');
INSERT INTO Performer (pid, pname, pphone) VALUES (0005, 'Luke Combs',
'2264474812');
INSERT INTO Performer (pid, pname, pphone) VALUES (0006, 'Justin Bieber',
'3068592277');
INSERT INTO Performer (pid, pname, pphone) VALUES (0007, 'Taylor Swift',
'2044825811');

-- Insert values into Attend_Locate table
INSERT INTO Attend_Locate(cid, pid, postalcode, street, buildingnumber)
VALUES(0001, 0001, 'V6T 1Z3', 'Thunderbird Blvd', 6066);
INSERT INTO Attend_Locate(cid, pid, postalcode, street, buildingnumber)
VALUES(0002, 0002, 'V5K 0A3', 'Griffiths Wy', 800);
INSERT INTO Attend_Locate(cid, pid, postalcode, street, buildingnumber)
VALUES(0003, 0003, 'T5J 0H6', '104 Ave NW', 10220);
INSERT INTO Attend_Locate(cid, pid, postalcode, street, buildingnumber)
VALUES(0004, 0004, 'H3B 5EB', 'Av. des Canadiens-de-Montreal', 1909);
INSERT INTO Attend_Locate(cid, pid, postalcode, street, buildingnumber)
VALUES(0005, 0005, 'M5J 2X2', 'Bay St', 40);

-- Insert values into AssistantAssist table
INSERT INTO AssistantAssist (aid, assistantname, assistantphone, pid)
VALUES (0001, 'David Liny', '2044703073', 0001);
INSERT INTO AssistantAssist (aid, assistantname, assistantphone, pid)
VALUES (0002, 'Dior James', '2498033823', 0004);
INSERT INTO AssistantAssist (aid, assistantname, assistantphone, pid)
VALUES (0003, 'Marry Potter', '3547795531', 0005);
```

```
INSERT INTO AssistantAssist (aid, assistantname, assistantphone, pid)
VALUES (0004, 'Anthony Brown', '4033886434', 0006);
INSERT INTO AssistantAssist (aid, assistantname, assistantphone, pid)
VALUES (0005, 'Nick Beal', '2634931237', 0007);
```