University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Conceptual and Logical Design

Date: 2018-10-14

Group Member:

NAME:	STUDENT ID:	CS ID:	TUTORIAL SECTION:	EMAIL:
KYO TANG	35163104	u0q0b	T1D	kyo.hideki.tang@gmail.com
CHRISTPHER YAO	20924163	d1e1b	T1D	yaowongzhou@gmail.com
RUI ZHANG	28834166	n3v0b	T1D	zhangrui_ca@sina.com
YANYUN BU	42828146	b6h1b	T1C	iamclaudebu@gmail.com

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 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

Schemas:

Form Explanation: The functional dependencies in all table will contains primary keys. The LHS for all functional dependencies are keys. Thus, all tables involving functional dependencies are 3NF or BCNF.

1. **Developer**

   Definition:

   Developer (accountNo:integer, password:string, name:string)

   Primary Key: accountNo

   Constraints: None

   FDs:

   accountNo->password
   accountNo->name

2. **Test Driver**

   Definition:

   TestDriver **(accountNo:integer**, driverid:string, phoneNo:integer)

   Primary Key: (accountNo,driverid)

   Foreign Key: accountNo references Developer

   Constraints: Weak Entity. Each Test Driver must be a developer.

   FDs:

   accountNo,driverlicense-> phoneNo

3. **Car**

   Definition:

   Car (carid:integer, cartype:string, **deviceid:integer**)

   Primary Key: carid

   Foreign Key: deviceid references SelfDrivingDevice

   Constraints: The relationship between Car and Self Driver Derive is one-to-one.

   FDs:

   carid->cartype
   carid->deviceid

4. **Self-Driving Device**

Definition:

SelfDrivingDevice (deviceid:integer, deviceVersion:string, **carid:intege**r)

Primary Key: deviceid

Foreign Key: carid references Car

Constraints: The relationship between Car and Self Driver Derive is one-to-one.

FDs:

deviceid->deviceVersion
deviceid->carid

5. **Self-Driving Software**

Definition:

SelfDrivingSoftware (versionid:integer, updatetime:string, comment:string)

Primary Key: versionid

FDs:

versionid->updatetime
versionid->comment

6. **The list of Software in Car**

Definition:

ListSoftwareInCar (**carid:integer, versionid:integer**)

Primary Key: (carid, versionid)

Foreign Key:
carid references Car
versionid references SelfDrivingSoftware

7. **Self-Driving Software Record**

Definition:

SelfDrivingSoftwareRecord (swrecordid:integer, consolelog:string, **versionid:integer**)

Primary Key: swrecordid

Foreign Key: versionid references Self Driving Software

Constraints: The relationship between Self-Driving Software Record and Self-Driving Software is many-to-one

FDs:

swrecordid->consolelog
swrecordid->versionid

8. **Self-Driving Test**

Definition:

SelfDrivingTest (recordid:integer, status:string, **carid:integer, versionid:integer, swrecordid:integer, pathid:integer, driverid:integer**, fromdatetime:string, todatetime:string)

Primary Key: recordid

Foreign Key:
carid references Car
versionid references SelfDrivingSoftware
swrecordid references Self Driving Software Record
pathid references Path
driverid references TestDriver

FDs:

recordid->status
recordid->carid
recordid->versionid
recordid->swrecordid
recordid->pathid
recordid->driverid
recordid->fromtime
recordid->totime

9. Path

Definition:

Path (pathid:integer, city:string, location:string, startpoint:string, endpoint: string, **pathcondid:integer**)

Primary Key: pathid

Foreign Key:
Pathcondid references PathCondition

Constraints: The relationship between Path and Path Condition is many-to-one

FDs:

pathid->city
pathid->location
pathid->startpoint
pathid->endpoint
pathid->pathcondid

10. Path Condition

Definition:

PathCondition (pathcondid:integer, roadtype:string, weather:string, climate:string, dayornight:string)

Primary Key: pathcondid

FDs:

pathcondid->roadtype
pathcondid->weather
pathcondid->climate
pathcondid->dayornight

11. Geometry

Definition:

Geometry (geoid:integer, lat:integer, lon:integer, **pathid:integer**)

Primary Key: (geoid, pathid)

Foreign Key: pathid

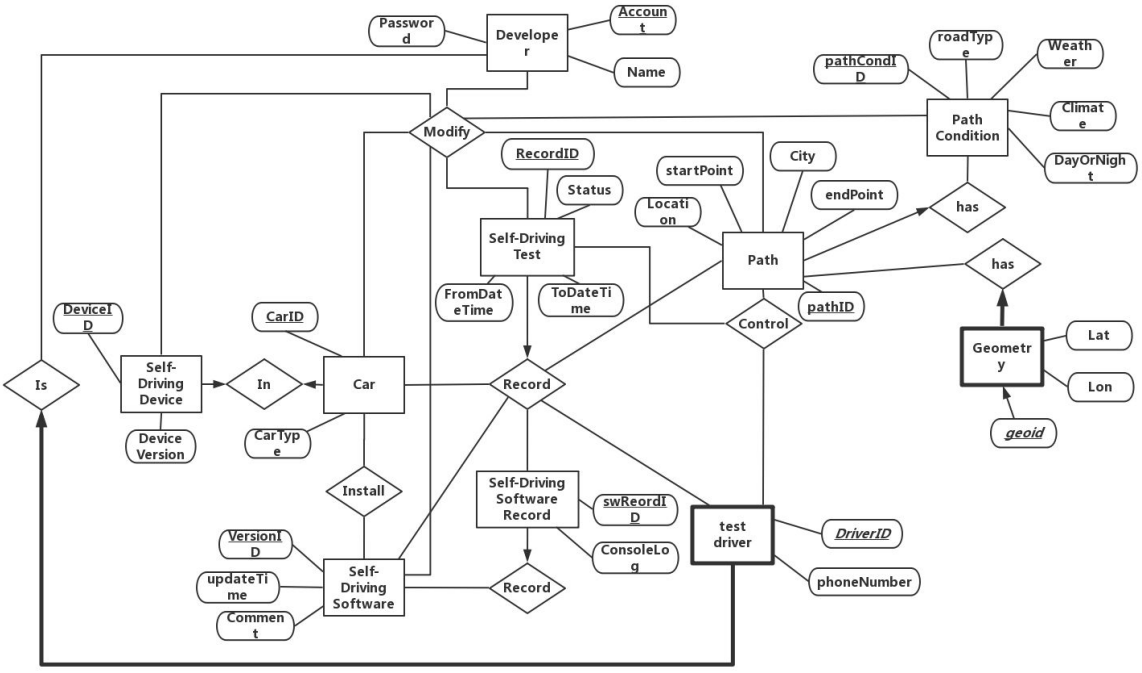
Constraints: Weak Entity. Each Geometry must be related with one Path.

FDs:

geoid, pathid->lat
geoid, pathid->lon

geoid, pathid->pathid

ER-diagram:



Note:

1.Self-Driving Software is an entity that stores a list of self-driving software in beta version that need be tested and improved on the research.
2.The italic and bold attributes are the key of a weak entiry.
3.Developer can create and modify Car, Self-Driving Device, Self-Driving Software, Self-Driving Software Record, Path, and Path Condition.
4.The geometry for each Path is generated by the other GPS software in our application depending on the startpoint and endpoint.
5.The Self-Driving Software Record is generated by Self-Driving Software.
6. Car must contains at least one Self-Driving Software
7. The relationship between Car and Self-Driving Software is many-to-many. Thus, we need one table(ListSoftwareInCar) to store this information.
8. Self-Driving Software may not be installed in any Car.
9. Self-Driving Test Record and Setting must has information for Car, Self-Driving Software, Path, and Test Driver.
10. Self-Driving Test Record and Setting may not contain Self-Driving Software Record. It depends on status. If the status is “finished”, Self-Driving Test Record and Setting must contain Self-Driving Software Record.

SQL DDL :

1. Developer

```
CREATE TABLE Developer
        (accountNo INTEGER,
         password  INTEGER,
         name      CHAR(30),
```

PRIMARY KEY (accountNo))

2. Test Driver

```
CREATE TABLE Test Driver
        (accountNo INTEGER,
         phoneNo   INTEGER,
         driverID     CHAR(30),
         PRIMARY KEY (accountNO, driverID)
         FOREIGN  KEY  (accountNo) REFERENCES Developer
         ON DELETE CASCADE
         ON UPDATE CASCADE)
```

3. Car

```
CREATE TABLE Car
        (carID      INTEGER,
         cartype    CHAR(30),
         deviceID  INTEGER,
         PRIMARY KEY (carID),
         FOREIGN  KEY (deviceID) REFERENCES Self Driving Device
         ON DELETE SET DEFAULT
         ON UPDATE CASCADE)
```

4. Self-Driving Device

```
CREATE TABLE SelfDrivingDevice,
        (deviceID        INTEGER,
         deviceVersion CHAR(30),
         carID             INTEGER,
         PRIMARY KEY (deviceID),
         FOREIGN  KEY (carID) REFERENCES Car
         ON DELETE SET DEFAULT
         ON UPDATE CASCADE)
```

5. Self-Driving Software

```
CREATE TABLE SelfDrivingSoftware
        (versionID    INTEGER,
         updatetime CHAR(30),
         comment    CHAR(1000)
         PRIMARY KEY (versionID))
```

6. The list of Software in Car

```
CREATE TABLE ListSoftwareInCar
        (carID      INTEGER,
         versionID INTEGER,
         PRIMARY KEY (carID ,versionID),
         FOREIGN  KEY (carID) REFERENCES Car
```

ON DELETE CASCADE
ON UPDATE CASCADE
FOREIGN KEY (versionID) **REFERENCES** SelfDrivingSoftware
ON DELETE CASCADE
ON UPDATE CASCADE)

7. Self-Driving Software Record

CREATE TABLE SelfDrivingSoftwareRecord
(swrecordID **INTEGER**,
consolelog **CHAR(1000)**,
versionID **INTEGER**,
PRIMARY KEY (swrecordID),
FOREIGN KEY (versionID) **REFERENCES** SelfDrivingSoftware
ON DELETE SET DEFAULT
ON UPDATE CASCADE)

8. Self-Driving Test

CREATE TABLE Self Driving TestRecordAndSetting
(recordID **INTEGER**,
status **CHAR(30)**,
carID **INTEGER**,
versionID **INTEGER**,
swrecordID **INTEGER**,
pathID **INTEGER**,
driverID **INTEGER**,
fromdatetime **CHAR(30)**,
todatetime **CHAR(30)**,
PRIMARY KEY (recordID),
FOREIGN KEY (carID) **REFERENCES** Car
ON DELETE SET DEFAULT
ON UPDATE CASCADE
FOREIGN KEY (versionID) **REFERENCES** SelfDrivingSoftware
ON DELETE SET DEFAULT
ON UPDATE CASCADE
FOREIGN KEY (swrecordID) **REFERENCES** Self Driving Software Record
ON DELETE SET DEFAULT
ON UPDATE CASCADE
FOREIGN KEY (pathID) **REFERENCES** Path
ON DELETE SET DEFAULT
ON UPDATE CASCADE
FOREIGN KEY (driverID) **REFERENCES** TestDriver
ON DELETE SET DEFAULT
ON UPDATE CASCADE)

9. Path

```
CREATE TABLE Path
        (pathID      INTEGER,
         city        CHAR(30),
        location     CHAR(30),
        startpoint   CHAR(30),
        endpoint     CHAR(30),
        pathcondID   INTEGER,
        PRIMARY KEY (pathID),
        FOREIGN KEY (pathcondID) REFERENCES Path Condition
        ON DELETE SET DEFAULT
        ON UPDATE CASCADE)
```

10. Path Condtion

```
CREATE TABLE PathCondition
        (pathcondID INTEGER,
        PRIMARY KEY (pathcondID),
        roadtype   CHAR(30),
        weather    CHAR(30),
        climate    CHAR(30),
        dayornight CHAR(30))
```

11. Geometry

```
CREATE TABLE Geometry
        (geoID  INTEGER,
        lat     INTEGER,
        lon     INTEGER,
        pathID INTEGER,
        PRIMARY KEY (geoID, pathID),
        FOREIGN KEY (pathID) REFERENCES Path
        ON DELETE CASCADE
        ON UPDATE CASCADE)
```