

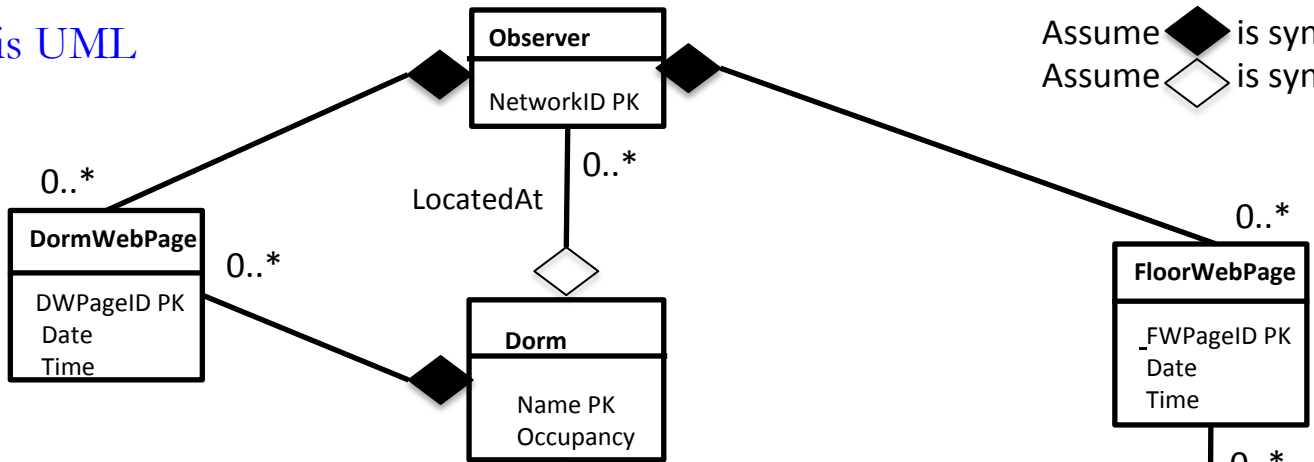
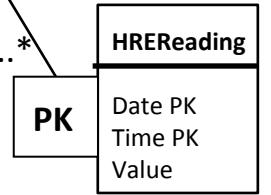
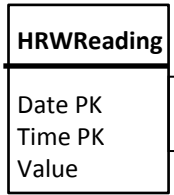
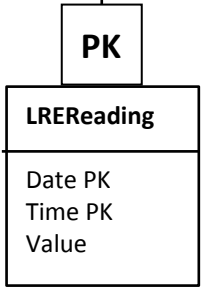
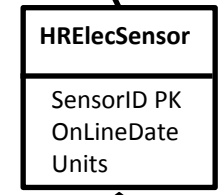
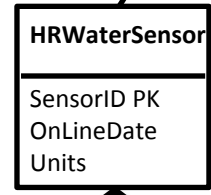
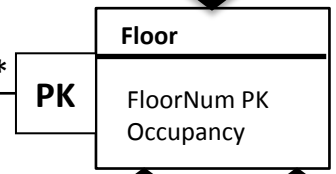
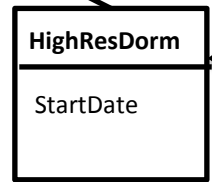
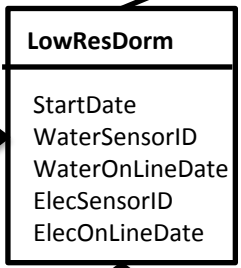
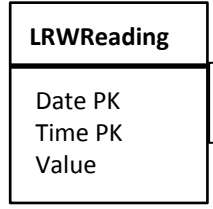
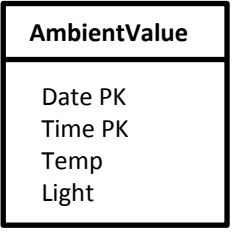


Translate this UML to SQL

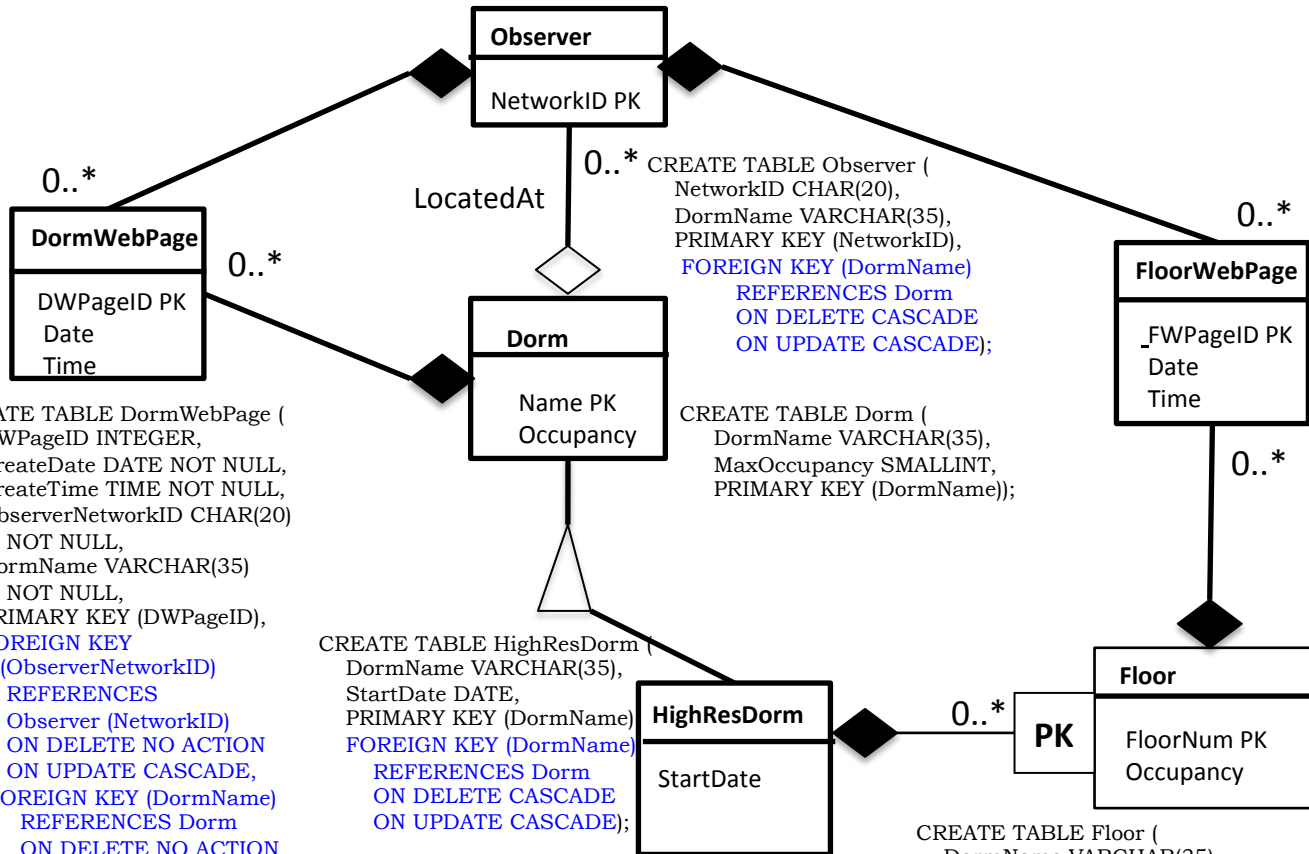
Assume  is synonymous to 1..1
 Assume  is synonymous to 0..1



*Complete coverage
No overlap*



This is one conceptual design for the Oberlin dorm energy monitoring DB. There are other possibilities. Post comments or questions about this design and alternative designs to the course discussion board.



```
CREATE TABLE Observer (
  NetworkID CHAR(20),
  DormName VARCHAR(35),
  PRIMARY KEY (NetworkID),
  FOREIGN KEY (DormName)
  REFERENCES Dorm
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE Dorm (
  DormName VARCHAR(35),
  MaxOccupancy SMALLINT,
  PRIMARY KEY (DormName));
```

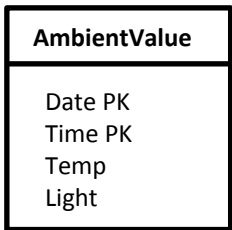
```
CREATE TABLE HighResDorm (
  DormName VARCHAR(35),
  StartDate DATE,
  PRIMARY KEY (DormName)
  FOREIGN KEY (DormName)
  REFERENCES Dorm
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE Floor (
  DormName VARCHAR(35),
  FloorNum TINYINT,
  MaxOccupancy SMALLINT,
  PRIMARY KEY (DormName, FloorNum),
  FOREIGN KEY (DormName)
  REFERENCES HighResDorm
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE FloorWebPage (
  DormName VARCHAR(35) NOT NULL,
  FloorNum TINYINT NOT NULL,
  FWPageID INTEGER,
  CreateDate DATE NOT NULL,
  CreateTime TIME NOT NULL,
  ObserverNetworkID CHAR(20) NOT NULL,
  PRIMARY KEY (FWPageID),
  FOREIGN KEY (ObserverNetworkID)
  REFERENCES Observer (NetworkID),
  ON DELETE NO ACTION
  ON UPDATE CASCADE,
  FOREIGN KEY (DormName, FloorNum)
  REFERENCES Floor
  ON DELETE NO ACTION
  ON UPDATE CASCADE );
```

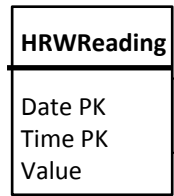
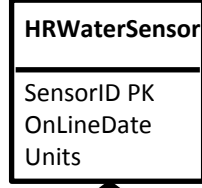
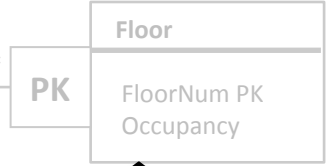
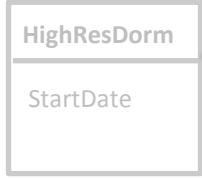
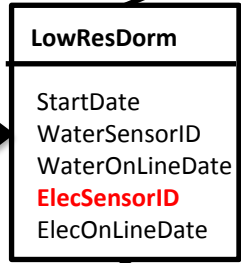
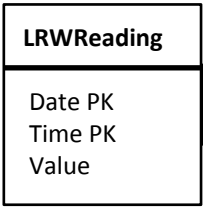
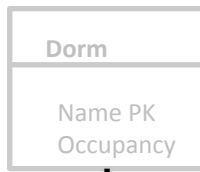
or RESTRICT

```
CREATE TABLE DormWebPage (
  DWPageID INTEGER,
  CreateDate DATE NOT NULL,
  CreateTime TIME NOT NULL,
  ObserverNetworkID CHAR(20)
  NOT NULL,
  DormName VARCHAR(35)
  NOT NULL,
  PRIMARY KEY (DWPageID),
  FOREIGN KEY
  (ObserverNetworkID)
  REFERENCES
  Observer (NetworkID)
  ON DELETE NO ACTION
  ON UPDATE CASCADE,
  FOREIGN KEY (DormName)
  REFERENCES Dorm
  ON DELETE NO ACTION
  ON UPDATE CASCADE );
```



```
CREATE TABLE AmbientValues (
  AmbientReadingsDate DATE,
  AmbientReadingsTime TIME,
  AmbientTemp TINYINT NOT NULL,
  AmbientLight CHAR(2),
  PRIMARY KEY (AmbientReadingsDate, AmbientReadingsTime));
```

```
CREATE TABLE LowResDorm (
  DormName VARCHAR(35),
  StartDate DATE,
  LRElecSensorID INTEGER NOT NULL,
  UNIQUE(LRElecSensorID),
  /*UNIQUE not inferable from UML */
  LRElecSensorOnLineDate DATE,
  LRWaterSensorOnLineDate DATE,
  LRWaterSensorID INTEGER,
  PRIMARY KEY (DormName),
  FOREIGN KEY (DormName) REFERENCES Dorm
  ON DELETE CASCADE ON UPDATE CASCADE );
```

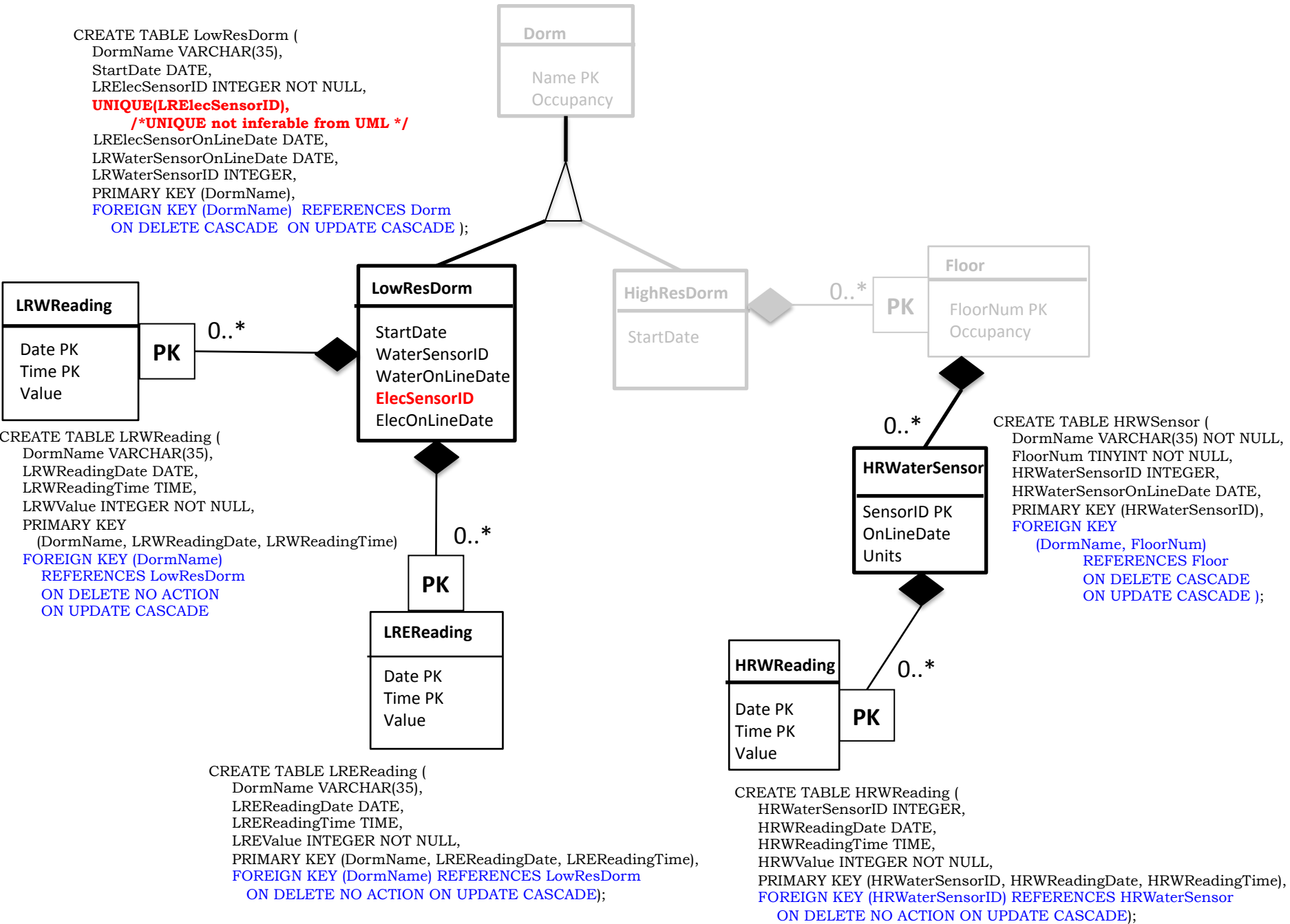


```
CREATE TABLE HRWSensor (
  DormName VARCHAR(35) NOT NULL,
  FloorNum TINYINT NOT NULL,
  HRWaterSensorID INTEGER,
  HRWaterSensorOnLineDate DATE,
  PRIMARY KEY (HRWaterSensorID),
  FOREIGN KEY
  (DormName, FloorNum)
  REFERENCES Floor
  ON DELETE CASCADE
  ON UPDATE CASCADE );
```

```
CREATE TABLE LRWReading (
  DormName VARCHAR(35),
  LRWReadingDate DATE,
  LRWReadingTime TIME,
  LRWValue INTEGER NOT NULL,
  PRIMARY KEY
  (DormName, LRWReadingDate, LRWReadingTime)
  FOREIGN KEY (DormName)
  REFERENCES LowResDorm
  ON DELETE NO ACTION
  ON UPDATE CASCADE
```

```
CREATE TABLE LREReading (
  DormName VARCHAR(35),
  LREReadingDate DATE,
  LREReadingTime TIME,
  LREValue INTEGER NOT NULL,
  PRIMARY KEY (DormName, LREReadingDate, LREReadingTime),
  FOREIGN KEY (DormName) REFERENCES LowResDorm
  ON DELETE NO ACTION ON UPDATE CASCADE);
```

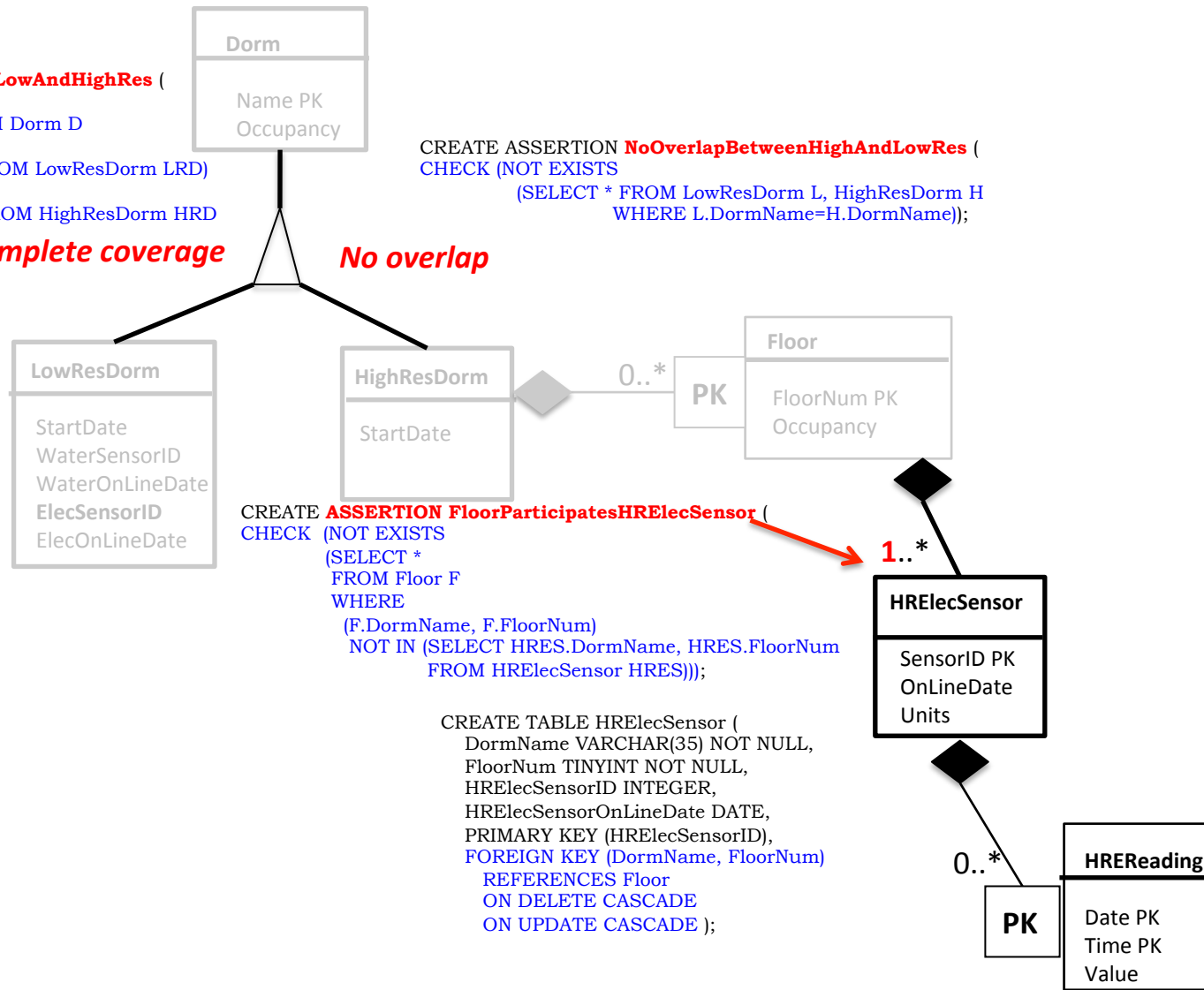
```
CREATE TABLE HRWReading (
  HRWaterSensorID INTEGER,
  HRWReadingDate DATE,
  HRWReadingTime TIME,
  HRWValue INTEGER NOT NULL,
  PRIMARY KEY (HRWaterSensorID, HRWReadingDate, HRWReadingTime),
  FOREIGN KEY (HRWaterSensorID) REFERENCES HRWaterSensor
  ON DELETE NO ACTION ON UPDATE CASCADE);
```



```
CREATE ASSERTION CompleteCoverOverLowAndHighRes (
  CHECK (NOT EXISTS
    (SELECT D.DormName FROM Dorm D
      EXCEPT
    SELECT LRD.DormName FROM LowResDorm LRD)
    EXCEPT
    SELECT HRD.DormName FROM HighResDorm HRD
  )
);
```

```
CREATE ASSERTION NoOverlapBetweenHighAndLowRes (
  CHECK (NOT EXISTS
    (SELECT * FROM LowResDorm L, HighResDorm H
      WHERE L.DormName=H.DormName));
```

Complete coverage **No overlap**



```
CREATE ASSERTION FloorParticipatesHRElecSensor (
  CHECK (NOT EXISTS
    (SELECT *
      FROM Floor F
      WHERE
        (F.DormName, F.FloorNum)
        NOT IN (SELECT HRES.DormName, HRES.FloorNum
          FROM HRElecSensor HRES)));
```

```
CREATE TABLE HRElecSensor (
  DormName VARCHAR(35) NOT NULL,
  FloorNum TINYINT NOT NULL,
  HRElecSensorID INTEGER,
  HRElecSensorOnLineDate DATE,
  PRIMARY KEY (HRElecSensorID),
  FOREIGN KEY (DormName, FloorNum)
  REFERENCES Floor
  ON DELETE CASCADE
  ON UPDATE CASCADE);
```

```
CREATE TABLE HREReading (
  HRElecSensorID INTEGER,
  HREReadingDate DATE,
  HREReadingTime TIME,
  HREValue INTEGER NOT NULL,
  PRIMARY KEY (HRElecSensorID, HREReadingDate,
  HREReadingTime),
  FOREIGN KEY (HRElecSensorID) REFERENCES HRElecSensor
  ON DELETE NO ACTION
  ON UPDATE CASCADE);
```