```
/* Assignment 4: Schema creation and constraints */

/*
      Modify the trains schema which we saw earlier  to create constraints to check the following:
*/

/*
      The value of timein is always less than or equal to timeout
*/

DELIMITER //

CREATE TRIGGER check_timeout_i BEFORE INSERT ON trainhalts FOR EACH ROW
BEGIN
      IF NEW.timein > NEW.timeout
      then signal sqlstate '22003'
      set message_text = 'timein should be less than timeout';
      END IF;
END; //

DELIMITER //

CREATE TRIGGER check_timeout_u BEFORE UPDATE ON trainhalts FOR EACH ROW
BEGIN
      IF NEW.timein > NEW.timeout
      then signal sqlstate '22003'
      set message_text = 'timein should be less than timeout';
      END IF;
END; //



/*
      When a train is removed from service, all its halts should be deleted.
*/

alter table trainhalts add foreign key (id) references train (id) on delete cascade on update cascade;



/*
      Insert inconsistent data and verify the constraints.
*/

insert into rail.trainhalts values ('A65', 10, 'CST', '20', '15');

/*
      OUTPUT:-
      ERROR 1644 (22003): timein should be less than timeout
*/




/*
      Write SQL Create table statements to create the following schema.
      Include all appropriate primary and foreign key declarations.
      Choose appropriate types for each attribute.
            remotecentre(centreId, college, town, state)
            person(ID, name, email)
            programme(progId, title, fromdate, todate)
            coordinator(ID, progId, centreId)
            participant(ID, progId, centreId)
*/
```

```sql
drop database prog;
create database prog;

create table remotecentre
      (centreId         numeric(5)        not null,
       college          varchar(20),
       town             varchar(10),
       state            varchar(10),
       primary key (centreID)
      );

create table person
      (ID               numeric(8)        not null,
       name             varchar(10),
       email            varchar(30),
       primary key (ID)
      );

create table programme
      (progId           numeric(4)        not null,
       title            varchar(30),
       fromdate         date,
       todate           date,
       primary key (progId)
      );

create table coordinator
      (ID               numeric(8),
       progId           numeric(4),
       centreId         numeric(5),
       primary key (ID, progId, centreId),
       foreign key (ID) references person (ID) on delete cascade on update cascade,
       foreign key (progId) references programme (progId) on delete cascade on update cascade,
       foreign key (centreId) references remotecentre (centreId) on delete cascade on update cascade
      );

create table participant
      (ID               numeric(8),
       progId           numeric(4),
       centreId         numeric(5),
       primary key (ID, progId, centreId),
       foreign key (ID) references person (ID) on delete cascade on update cascade,
       foreign key (progId) references programme (progId) on delete cascade on update cascade,
       foreign key (centreId) references remotecentre (centreId) on delete cascade on update cascade
      );
```