1.

Create or replace function errorreport() returns trigger as $checksession$

Begin

If (Exists (select \* from session where section\_id = NEW.section\_id and ((begintime >NEW.begintime and begintime < NEW.endtime) or (endtime >NEW.begintime and endtime < NEW.endtime) or (begintime <= NEW.begintime and endtime >= NEW.endtime)) and (position(weekday in NEW.weekday) > 0 or position(NEW.weekday in weekday) > 0))) then

Raise exception 'Error: sessions cant happen at same time';

End if;

Return NEW;

End

$checksession$ LANGUAGE plpgsql;

Create trigger checksession before insert or update on session

For each row Execute procedure errorreport();

2.

Create or replace function if\_full() returns trigger as $checkenrolllist$

Begin

If ((select count(\*) from enrolllist where sec\_id = NEW.sec\_id) = (select en\_limit from section where id = NEW.sec\_id)) then

Raise exception 'section is full';

End if;

Return NEW;

End

$checkenrolllist$ LANGUAGE plpgsql;

Create trigger checkenrolllist before insert on enrolllist

For each row Execute procedure if\_full();

3.

Create or replace function sectioncheck () returns trigger as $checkteach$

Begin

If (exists(select s1.\*, s2.\* from section s1, section s2 where s1.faculty = s2.faculty and s1.id != s2.id and exists(select \* from session m1, session m2 where m1.id != m2.id and (m1.section\_id = s1.id and m2.section\_id = s2.id and ((m1.begintime >m2.begintime and m1.begintime < m2.endtime) or (m1.endtime >m2.begintime and m1.endtime < m2.endtime) or (m1.begintime <= m2.begintime and m1.endtime >= m2.endtime))) and (position(m2.weekday in m1.weekday) > 0 or position(m1.weekday in m2.weekday) > 0)))) then

Delete from session where id = NEW.id;

Raise exception 'cant teach overlapping sections';

End if;

Return NEW;

End

$checkteach$ LANGUAGE plpgsql;

Create trigger checkteach after insert on session

For each row Execute procedure sectioncheck();