**Milestone 5**

1.

Create table CPQG (

Course\_id int not null,

Faculty varchar(50) not null,

Quarter varchar(50) not null,

Grade varchar(10) not null,

Count int,

Primary key(course\_id, faculty, quarter, grade)

);

Insert into CPQG (select co.id, s.faculty, cl.quarter, l.grade, count(\*) from course co, class cl, section s, learning l where l.section\_id = s.id and s.class = cl.id and cl.course\_id = co.id group by co.id, s.faculty, cl.quarter, l.grade);

2.

Create table CPG(

Course\_id int not null,

Faculty varchar(50) not null,

Grade varchar(10) not null,

Count int,

Primary key(course\_id, faculty, grade)

);

Insert into CPG (select co.id, s.faculty, l.grade, count(\*) from course co, class cl, section s, learning l where l.section\_id = s.id and s.class = cl.id and cl.course\_id = co.id group by s.faculty, co.id, l.grade);

3.

(1)CPQG update

Create or replace function CPQGupdate() returns trigger as $CPQGtrigger$

Begin

If (TG\_OP = 'UPDATE') then

update CPQG set count = count - 1 where grade = OLD.grade and exists (select co.id from course co, class cl, section s where OLD.section\_id = s.id and s.faculty = cpqg.faculty and s.class = cl.id and cl.quarter = cpqg.quarter and cl.course\_id = co.id and co.id = cpqg.course\_id);

Update CPQG set count = count + 1 where grade = NEW.grade and exists(select co.id from course co, class cl, section s where NEW.section\_id = s.id and s.faculty = cpqg.faculty and s.class = cl.id and cl.quarter = cpqg.quarter and cl.course\_id = co.id and co.id = cpqg.course\_id);

Elsif (TG\_OP = 'INSERT') then

If (not Exists(select co.id from course co, class cl, section s, CPQG cp where cp.grade = NEW.grade and NEW.section\_id = s.id and s.faculty = cp.faculty and s.class = cl.id and cl.quarter = cp.quarter and cl.course\_id = co.id and co.id = cp.course\_id)) then

Insert into CPQG (select co.id, s.faculty, cl.quarter, le.grade, 0 from course co, class cl, section s, learning le where le.student\_SSN = NEW.student\_SSN and le.section\_id = NEW.section\_id and le.section\_id = s.id and s.class = cl.id and cl.course\_id = co.id);

End if;

Update CPQG set count = count + 1 where grade = NEW.grade and exists(select co.id from course co, class cl, section s where NEW.section\_id = s.id and s.faculty = cpqg.faculty and s.class = cl.id and cl.quarter = cpqg.quarter and cl.course\_id = co.id and co.id = cpqg.course\_id);

End if;

Return NEW;

End

$CPQGtrigger$ LANGUAGE plpgsql;

Create trigger CPQGtrigger after insert or update on learning

For each row execute procedure CPQGupdate();

(2)CPG update

Create or replace function CPGupdate() returns trigger as $CPGtrigger$

Begin

If (TG\_OP = 'UPDATE') then

update CPG set count = count - 1 where grade = OLD.grade and exists (select co.id from course co, class cl, section s where OLD.section\_id = s.id and s.faculty = cpg.faculty and s.class = cl.id and cl.course\_id = co.id and co.id = cpg.course\_id);

Update CPG set count = count + 1 where grade = NEW.grade and exists(select co.id from course co, class cl, section s where NEW.section\_id = s.id and s.faculty = cpg.faculty and s.class = cl.id and cl.course\_id = co.id and co.id = cpg.course\_id);

Elsif (TG\_OP = 'INSERT') then

If (not Exists(select co.id from course co, class cl, section s, CPG cp where cp.grade = NEW.grade and NEW.section\_id = s.id and s.faculty = cp.faculty and s.class = cl.id and cl.course\_id = co.id and co.id = cp.course\_id)) then

Insert into CPG (select co.id, s.faculty, le.grade, 0 from course co, class cl, section s, learning le where le.student\_SSN = NEW.student\_SSN and le.section\_id = NEW.section\_id and le.section\_id = s.id and s.class = cl.id and cl.course\_id = co.id);

End if;

Update CPG set count = count + 1 where grade = NEW.grade and exists(select co.id from course co, class cl, section s where NEW.section\_id = s.id and s.faculty = cpg.faculty and s.class = cl.id and cl.course\_id = co.id and co.id = cpg.course\_id);

End if;

Return NEW;

End

$CPGtrigger$ LANGUAGE plpgsql;

Create trigger CPGtrigger after insert or update on learning

For each row execute procedure CPGupdate();