/\* USER\_PROFILE owner must be a USER \*/

create or replace TRIGGER Profile\_Assertion\_For\_User\_Profile

AFTER INSERT OR UPDATE ON USER\_PROFILE

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_USER,CREATOR\_PROFILE,CREATOR\_HAS\_PROFILE

WHERE :NEW.ProfileId=CREATOR\_PROFILE.ProfileId AND

CREATOR\_PROFILE.ProfileId=CREATOR\_HAS\_PROFILE.ProfileId AND

CREATOR\_HAS\_PROFILE.CreatorId=SOCIAL\_RESUME\_USER.CreatorId;

IF user\_count = 0 THEN

raise\_application\_error (-20000, 'USER\_PROFILE owner must be a SOCIAL\_RESUME\_USER');

END IF;

END;

/

/\*COMPANY\_PROFILE owner must be a COMPANY \*/

create or replace TRIGGER Profile\_Assertion\_For\_Company\_Profile

AFTER INSERT OR UPDATE ON COMPANY\_PROFILE

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY,CREATOR\_PROFILE,CREATOR\_HAS\_PROFILE

WHERE :NEW.ProfileId=CREATOR\_PROFILE.ProfileId AND

CREATOR\_PROFILE.ProfileId=CREATOR\_HAS\_PROFILE.ProfileId AND

CREATOR\_HAS\_PROFILE.CreatorId=COMPANY.CreatorId;

IF user\_count = 0 THEN

raise\_application\_error (-20001, 'COMPANY\_PROFILE owner must be a COMPANY');

END IF;

END;

/

/\*PAGE\_PROFILE owner must be a PAGE \*/

create or replace TRIGGER Profile\_Assertion\_For\_Page\_Profile

AFTER INSERT OR UPDATE ON PAGE\_PROFILE

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_PAGE,CREATOR\_PROFILE,CREATOR\_HAS\_PROFILE

WHERE :NEW.ProfileId=CREATOR\_PROFILE.ProfileId AND

CREATOR\_PROFILE.ProfileId=CREATOR\_HAS\_PROFILE.ProfileId AND

CREATOR\_HAS\_PROFILE.CreatorId=SOCIAL\_RESUME\_PAGE.CreatorId;

IF user\_count = 0 THEN

raise\_application\_error (-20002, 'PAGE\_PROFILE owner must be a SOCIAL\_RESUME\_PAGE');

END IF;

END;

/

/\*JOB owner must not be an USER\*/

create or replace TRIGGER Action\_Job\_For\_User AFTER INSERT OR UPDATE ON JOB\_ANNOUNCEMENT

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_USER

WHERE :NEW.CreatorId=SOCIAL\_RESUME\_USER.CreatorId;

IF user\_count <> 0 THEN

raise\_application\_error (-20003, 'JOB owner must not be an SOCIAL\_RESUME\_USER');

END IF;

END;

/

/\*FOLLOW owner can not be a COMPANY\*/

create or replace TRIGGER Action\_Follow\_For\_Company AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY,COMMUNITY

WHERE COMPANY.CommunityId=COMMUNITY.CommunityId

AND :NEW.CreatorId=COMPANY.CreatorId

AND :NEW.ActionType='FOLLOW';

IF user\_count <> 0 THEN

raise\_application\_error (-20004, 'FOLLOW owner can not be a COMPANY');

END IF;

END;

/

/\*JOIN owner can not be a COMPANY\*/

create or replace TRIGGER Action\_Join\_For\_Company AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY,COMMUNITY

WHERE COMPANY.CommunityId=COMMUNITY.CommunityId

AND :NEW.CreatorId=COMPANY.CreatorId

AND :NEW.ActionType='JOIN';

IF user\_count <> 0 THEN

raise\_application\_error (-20005, 'JOIN owner can not be a COMPANY');

END IF;

END;

/

/\*CONNECTION owner can not be a COMPANY\*/

create or replace TRIGGER Action\_Connection\_For\_Company AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY,COMMUNITY

WHERE COMPANY.CommunityId=COMMUNITY.CommunityId

AND :NEW.CreatorId=COMPANY.CreatorId

AND :NEW.ActionType='CONNECTION';

IF user\_count <> 0 THEN

raise\_application\_error (-20006, 'CONNECTION owner can not be a COMPANY');

END IF;

END;

/

/\*CONNECTION owner can not be a PAGE\*/

create or replace TRIGGER Action\_Connection\_For\_Page AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_PAGE,COMMUNITY

WHERE SOCIAL\_RESUME\_PAGE.CommunityId=COMMUNITY.CommunityId

AND :NEW.CreatorId=SOCIAL\_RESUME\_PAGE.CreatorId

AND :NEW.ActionType='CONNECTION';

IF user\_count <> 0 THEN

raise\_application\_error (-20007, 'CONNECTION owner can not be a PAGE');

END IF;

END;

/

/\*JOIN owner can not be a PAGE\*/

create or replace TRIGGER Action\_Join\_For\_Page AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_PAGE,COMMUNITY

WHERE SOCIAL\_RESUME\_PAGE.CommunityId=COMMUNITY.CommunityId

AND :NEW.CreatorId=SOCIAL\_RESUME\_PAGE.CreatorId

AND :NEW.ActionType='JOIN';

IF user\_count <> 0 THEN

raise\_application\_error (-20008, 'JOIN owner can not be a PAGE');

END IF;

END;

/

/\*CREATOR\_PROFILE can not be affected with Comment action\*/

create or replace TRIGGER Effected\_Action\_Comment\_For\_Creator\_Profile

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM CREATOR\_PROFILE,CREATOR

WHERE :NEW.ActionType='COMMENT' AND

CREATOR\_PROFILE.EffectedId= :NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20009, 'CREATOR\_PROFILE can not be affected with COMMENT action');

END IF;

END;

/

/\*CREATOR\_PROFILE can not be affected with LIKE action\*/

create or replace TRIGGER Effected\_Action\_Like\_For\_Creator\_Profile

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM CREATOR\_PROFILE,CREATOR

WHERE :NEW.ActionType='LIKE' AND

CREATOR\_PROFILE.EffectedId= :NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20010, 'CREATOR\_PROFILE can not be affected with LIKE action');

END IF;

END;

/

/\*CREATOR\_PROFILE can not be affected with JOIN action\*/

create or replace TRIGGER Effected\_Action\_Join\_For\_Creator\_Profile

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM CREATOR\_PROFILE,CREATOR

WHERE :NEW.ActionType='JOIN' AND

CREATOR\_PROFILE.EffectedId= :NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20011, 'CREATOR\_PROFILE can not be affected with JOIN action');

END IF;

END;

/

/\*COMPANY\_PROFILE can not be affected with CONNECTION action\*/

create or replace TRIGGER Effected\_Action\_Connection\_For\_Company\_Profile AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY\_PROFILE,CREATOR\_PROFILE

WHERE :NEW.ActionType='CONNECTION' AND

COMPANY\_PROFILE.ProfileId=CREATOR\_PROFILE.ProfileId AND

CREATOR\_PROFILE.ProfileId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20012, 'CREATOR\_PROFILE can not be affected with CONNECTION action');

END IF;

END;

/

/\*PAGE\_PROFILE can not be affected with CONNECTION action\*/

create or replace TRIGGER Effected\_Action\_Connection\_For\_Page\_Profile

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM PAGE\_PROFILE,CREATOR\_PROFILE

WHERE :NEW.ActionType='CONNECTION' AND

PAGE\_PROFILE.ProfileId=CREATOR\_PROFILE.ProfileId AND

CREATOR\_PROFILE.ProfileId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20013, 'PAGE\_PROFILE can not be affected with CONNECTION action');

END IF;

END;

/

/\*GROUP can not be affected with Search action\*/

create or replace TRIGGER Effected\_Action\_Search\_For\_Group AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='SEARCH' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20014, 'GROUP can not be affected with SEARCH action');

END IF;

END;

/

/\*GROUP can not be affected with VIEW action\*/

create or replace TRIGGER Effected\_Action\_View\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='VIEW' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20015, 'GROUP can not be affected with VIEW action');

END IF;

END;

/

/\*GROUP can not be affected with Message action\*/

create or replace TRIGGER Effected\_Action\_Message\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='MESSAGE' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20016,'GROUP can not be affected with MESSAGE action');

END IF;

END;

/

/\*GROUP can not be affected with Like action\*/

create or replace TRIGGER Effected\_Action\_Like\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='LIKE' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error(-20017, 'GROUP can not be affected with LIKE action');

END IF;

END;

/

/\*GROUP can not be affected with Connection action\*/

create or replace TRIGGER Effected\_Action\_Connection\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='CONNECTION' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20018, 'GROUP can not be affected with CONNECTION action');

END IF;

END;

/

/\*GROUP can not be affected with Follow action\*/

create or replace TRIGGER Effected\_Action\_Follow\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='FOLLOW' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20019, 'GROUP can not be affected with FOLLOW action');

END IF;

END;

/

/\*GROUP can not be affected with Comment action\*/

create or replace TRIGGER Effected\_Action\_Comment\_For\_Group

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM SOCIAL\_RESUME\_GROUP

WHERE :NEW.ActionType='COMMENT' AND

SOCIAL\_RESUME\_GROUP.EffectedId=:NEW.DestinationId;

IF user\_count <> 0 THEN

raise\_application\_error (-20020, 'GROUP can not be affected with COMMENT action');

END IF;

END;

/

/\*COMPANY can not be affected with Join action\*/

create or replace TRIGGER Effected\_Action\_Join\_For\_Company

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM COMPANY,CREATOR,CREATOR\_HAS\_PROFILE,CREATOR\_PROFILE

WHERE COMPANY.CreatorId=CREATOR.CreatorId AND

CREATOR\_HAS\_PROFILE.CreatorId=CREATOR.CreatorId AND

CREATOR\_PROFILE.ProfileId=CREATOR\_HAS\_PROFILE.ProfileId AND

CREATOR\_PROFILE.EffectedId=:NEW.DestinationId AND

:NEW.ActionType='JOIN';

IF user\_count <> 0 THEN

raise\_application\_error (-20021, 'COMPANY can not be affected with JOIN action');

END IF;

END;

/

/\*POST can not be affected with Search action\*/

create or replace TRIGGER Effected\_Action\_Search\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='SEARCH' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20022, 'POST can not be affected with SEARCH action');

END IF;

END;

/

/\*POST can not be affected with View action\*/

create or replace TRIGGER Effected\_Action\_View\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='VIEW' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20023, 'POST can not be affected with VIEW action');

END IF;

END;

/

/\*POST can not be affected with Message action\*/

create or replace TRIGGER Effected\_Action\_Message\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='MESSAGE' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20024, 'POST can not be affected with MESSAGE action');

END IF;

END;

/

/\*POST can not be affected with Follow action\*/

create or replace TRIGGER Effected\_Action\_Follow\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='FOLLOW' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20025, 'POST can not be affected with FOLLOW action');

END IF;

END;

/

/\*POST can not be affected with Join action\*/

create or replace TRIGGER Effected\_Action\_Join\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='JOIN' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20026, 'POST can not be affected with JOIN action');

END IF;

END;

/

/\*POST can not be affected with Connection action\*/

create or replace TRIGGER Effected\_Action\_Connection\_For\_Post AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='CONNECTION' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20027, 'POST can not be affected with CONNECTION action');

END IF;

END;

/

/\*POST can not be affected with Post action\*/

create or replace TRIGGER Effected\_Action\_Post\_For\_Post

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT (\*) INTO user\_count

FROM ACTION\_SHARE

WHERE ACTION\_SHARE.ShareType='POST' AND

:NEW.ActionType='POST' AND

:NEW.DestinationId=ACTION\_SHARE.EffectedId;

IF user\_count <> 0 THEN

raise\_application\_error (-20028, 'POST can not be affected with POST action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Search action\*/

create or replace TRIGGER Effected\_Action\_Search\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='SEARCH';

IF user\_count <> 0 THEN

raise\_application\_error (-20029, 'CREATOR\_COMMENT can not be affected with SEARCH action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with View action\*/

create or replace TRIGGER Effected\_Action\_View\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='VIEW';

IF user\_count <> 0 THEN

raise\_application\_error (-20030, 'CREATOR\_COMMENT can not be affected with VIEW action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Message action\*/

create or replace TRIGGER Effected\_Action\_Message\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='MESSAGE';

IF user\_count <> 0 THEN

raise\_application\_error (-20031, 'CREATOR\_COMMENT can not be affected with MESSAGE action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Follow action\*/

create or replace TRIGGER Effected\_Action\_Follow\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='FOLLOW';

IF user\_count <> 0 THEN

raise\_application\_error (-20032, 'CREATOR\_COMMENT can not be affected with FOLLOW action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Join action\*/

create or replace TRIGGER Effected\_Action\_Join\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='JOIN';

IF user\_count <> 0 THEN

raise\_application\_error (-20033, 'CREATOR\_COMMENT can not be affected with JOIN action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Connection action\*/

create or replace TRIGGER Effected\_Action\_Connection\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='CONNECTION';

IF user\_count <> 0 THEN

raise\_application\_error (-20034, 'CREATOR\_COMMENT can not be affected with CONNECTION action');

END IF;

END;

/

/\*CREATOR\_COMMENT can not be affected with Post action\*/

create or replace TRIGGER Effected\_Action\_Post\_For\_Creator\_Comment

AFTER INSERT OR UPDATE ON ACTION

FOR EACH ROW

DECLARE user\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO user\_count

FROM CREATOR\_COMMENT

WHERE CREATOR\_COMMENT.EffectedId= :NEW.DestinationId AND

:NEW.ActionType='POST';

IF user\_count <> 0 THEN

raise\_application\_error (-20035, 'CREATOR\_COMMENT can not be affected with POST action');

END IF;

END;

/