**Álgebra relacional**:

USUARIO (dni, email, nombre)

ACTIVIDAD (#actividad, #proyecto, nombre, detalle\_actividad, cond\_actividad)

PROYECTO (#proyecto, nombre, presupuesto, cantidad\_días)

COMENTARIO (#comentario, dni, #actividad, valoración, detalle\_comentario)

- Hallar el #proyecto y nombre de los proyectos en los que todas sus actividades tienen al menos 1 comentario.

ACT\_1\_COMENT = πactividad,proyecto (ACTIVIDAD ⨝ COMENTARIO)

ACT\_SIN\_COMENT = πactividad,proyecto (ACTIVIDAD) - ACT\_1\_COMENT

πproyecto(ACTIVIDAD) - πproyecto (ACT\_SIN\_COMENT⨝PROYECTO)

1° -> MAL:

ACTIVIDAD\_AL\_MENOS\_1\_COMENT <- (ACTIVIDAD |X| COMENTARIO)

PROY\_ACT\_1\_COMENT <- π(#proyecto, nombre, #actividad) (ACTIVIDAD\_AL\_MENOS\_1\_COMENT |X| PROYECTO)

π(#proyecto, nombre) (PROY\_ACT\_1\_COMENT % π(#actividad) ACTIVIDAD)

2° -> MAL:

PROY\_ACT\_1\_COMENT <- π(#proyecto, #actividad) (ACTIVIDAD |X| COMENTARIO)

PROY\_ACT\_SIN\_COMENT <- π(#proyecto, #actividad) (ACTIVIDAD) - PROY\_ACT\_1\_COMENT

PROY\_FINAL <- π(#proyecto) (π(#proyecto, #actividad) (ACTIVIDAD) - PROY\_ACT\_SIN\_COMENT)

π(#proyecto, nombre) (PROY\_FINAL |X| PROYECTO)

**MySQL**:

**1)** USUARIO (dni, email, nombre)

ACTIVIDAD (#actividad, #proyecto, nombre, detalle\_actividad, cond\_actividad, cant\_comentarios)

PROYECTO (#proyecto, nombre, presupuesto, cantidad\_días)

COMENTARIO (#comentario, dni, #actividad, valoración, detalle\_comentario)

Trigger para que con cada nuevo comentario registrado se actualice cant\_comentarios:

DELIMITER //

CREATE TRIGGER addCantidadComentarios

AFTER INSERT ON comentario FOR EACH ROW

BEGIN

UPDATE actividad

SET cant\_comentarios = cant\_comentarios + 1

WHERE #actividad = NEW.#actividad;

END;

// DELIMITER ;

**2)** Explicar 2 motivos por los cuales necesitaría definir 1 vista:

1. Seguridad -> restringir el acceso a los datos, controlando qué columnas o filas específicas se muestran a determinados usuarios, limitando el acceso a información sensible o irrelevante. Esto mejora la seguridad y el control de acceso a los datos.

2. Simplicidad -> hacer las consultas complejas más fáciles de usar para los usuarios, proporcionando a los usuarios una interfaz más sencilla y rápida para acceder a datos derivados de múltiples tablas o cálculos.