

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 = $\pi_{\text{actividad, proyecto}}(\text{ACTIVIDAD} \bowtie \text{COMENTARIO})$

ACT_SIN_COMENT = $\pi_{\text{actividad, proyecto}}(\text{ACTIVIDAD}) - \text{ACT_1_COMENT}$

$\pi_{\text{proyecto}}(\text{ACTIVIDAD}) - \pi_{\text{proyecto}}(\text{ACT_SIN_COMENT} \bowtie \text{PROYECTO})$

1° -> MAL:

ACTIVIDAD_AL_MENOS_1_COMENT <- (ACTIVIDAD |X| COMENTARIO)

PROY_ACT_1_COMENT <- $\pi(\#proyecto, nombre, \#actividad)$
(ACTIVIDAD_AL_MENOS_1_COMENT |X| PROYECTO)

$\pi(\#proyecto, nombre) (\text{PROY_ACT_1_COMENT} \% \pi(\#actividad) \text{ ACTIVIDAD})$

2° -> MAL:

PROY_ACT_1_COMENT <- $\pi(\#proyecto, \#actividad) (\text{ACTIVIDAD} |X| \text{COMENTARIO})$

PROY_ACT_SIN_COMENT <- $\pi(\#proyecto, \#actividad) (\text{ACTIVIDAD}) - \text{PROY_ACT_1_COMENT}$

PROY_FINAL <- $\pi(\#proyecto) (\pi(\#proyecto, \#actividad) (\text{ACTIVIDAD}) - \text{PROY_ACT_SIN_COMENT})$

$\pi(\#proyecto, nombre) (\text{PROY_FINAL} |X| \text{PROYECTO})$

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.