Transacciones - Ejercicios

1. Mostrar todas las tareas con su título, estado, prioridad y nombre del usuario que la creó.

```sql
SELECT
  t.title,
  t.status,
  p.p_name AS prioridad,
  u.name AS usuario
FROM tasks t
JOIN users u ON t.uid = u.uid
JOIN priorities p ON t.pid = p.pid;
```

tasks(+) 1

SELECT t.title, t.status, p.p_name AS pr | Enter a SQL expression to filter results (use Ctrl+Space)

	title	status	prioridad	usuario
1	Actualizar sitio web	pendiente	baja	María López
2	Planificar reunión	completada	media	Ana Torres
3	Redactar informe mensual	pendiente	media	Ana Torres
4	Revisar código fuente	en progreso	alta	Luis Pérez

2. Listar las tareas que tienen como estado "pendiente".

```sql
SELECT title, status
FROM tasks
WHERE status = 'pendiente';
```

tasks 1

SELECT title, status FROM tasks WHER | Enter a SQL expression to filter results (use Ctrl+Space)

	title	status
1	Redactar informe mensual	pendiente
2	Actualizar sitio web	pendiente

3. Ver cuántas tareas hay por cada nivel de prioridad.

```sql
SELECT p.p_name AS prioridad, COUNT(*) AS cantidad
FROM tasks t
JOIN priorities p ON t.pid = p.pid
GROUP BY p.p_name;
```

priorities 1

SELECT p.p_name AS prioridad, COUN | Enter a SQL expression to filter results (use Ctrl+Space)

	prioridad	cantidad
1	media	2
2	alta	1
3	baja	1

4. Mostrar los comentarios con el nombre del usuario y el título de la tarea a la que pertenecen.

```sql
SELECT
  c.summary AS comentario,
  u.name AS usuario,
  t.title AS tarea
FROM comments c
LEFT JOIN users u ON c.uid = u.uid
LEFT JOIN tasks t ON c.tid = t.tid;
```

comments(+) 1

SELECT c.summary AS comentario, u.r | Enter a SQL expression to filter results (use Ctrl+Space)

	comentario	usuario	tarea
1	Revisar cifras antes del viernes.	Ana Torres	Redactar informe mensual
2	Problemas encontrados en la validación del token.	Luis Pérez	Revisar código fuente
3	Agregar nuevos estilos CSS.	María López	Actualizar sitio web
4	Cliente confirmó la hora.	Ana Torres	Planificar reunión

5. Crear una vista task_high que muestre solo tareas con prioridad "alta".

```sql
CREATE VIEW task_high AS
SELECT t.tid, t.title, t.status, p.p_name AS prioridad
FROM tasks t
JOIN priorities p ON t.pid = p.pid
WHERE p.p_name = 'alta';
```

Estadísticas 1

Name	Value
Updated Rows	0
Execute time	0.101s
Start time	Mon May 19 00:37:22 COT 2025
Finish time	Mon May 19 00:37:22 COT 2025
Query	CREATE VIEW task_high AS
	SELECT t.tid, t.title, t.status, p.p_name AS prioridad
	FROM tasks t
	JOIN priorities p ON t.pid = p.pid
	WHERE p.p_name = 'alta'

6. Transferir tarea de un usuario a otro usuario.

```sql
BEGIN;

UPDATE tasks
SET uid = 6
WHERE tid = 26;

COMMIT;

SELECT tid, title, uid FROM tasks WHERE tid = 26;
```

Salida

Enter a part of a message to search for here

```
ya hay una transacción en curso
```

Estadísticas 1 | Estadísticas 2 | tasks 3

SELECT tid, title, uid FROM tasks WHE | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla / Texto

	tid	title	uid
1	26	Redactar informe mensual	6

Valor: 26

7. Clonar una tarea (tid = 27) incluyendo las etiquetas.

```sql
BEGIN;

INSERT INTO tasks (title, summary, status, created_date, limit_date, uid, pid)
SELECT
  title || ' (clonada)',
  summary,
  status,
  CURRENT_TIMESTAMP,
  limit_date,
  7,
  pid
FROM tasks
WHERE tid = 27
RETURNING tid;
```

Estadísticas 6 | tasks 7 | task_tags 7 (2) | Estadísticas 7 | task_tags(+) 8 | Estadísticas 8 | tasks 9

BEGIN; INSERT INTO tasks (title, sumr | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla

	tid
1	34

Valor: 34