DATABASE STAR

Legacy Database Analysis Guide

A reference guide to help you understand a legacy database

Step 1: Get the Big Picture

Find out what platform or vendor you are using.
Ask someone, or run this SQL.

```
--Works on Postgres and MySQL:
SELECT version();

--Works on SQL Server:
SELECT @@version;

--Works on Oracle:
SELECT * FROM v$version;
```

Find the number of tables

```
SELECT table_schema, count(*)
FROM information_schema.tables
GROUP BY table_schema;
```

Find the names of tables

```
SELECT table_schema, table_name
FROM information_schema.tables
WHERE table_schema = 'public'
ORDER BY table_schema ASC, table_name
ASC;
```

Find the number of views

```
SELECT COUNT(*) FROM
information_schema.views;
```

Find the number of functions or stored procedures

```
SELECT COUNT(*)
FROM information_schema.routines
WHERE routine_type IN ('FUNCTION',
'PROCEDURE')
AND specific_schema = 'public';
```

Step 2: Map the Structure

Create an ERD (Entity Relationship Diagram) to show tables, columns, and relationships.
Ideally, generate one from your SQL editor rather than creating manually

Step 3: Explore the Data

Start with SELECT queries on some tables:
Use SELECT * to see all columns
Limit the rows to a small number by using either:

- TOP 10
- LIMIT 10
- FETCH FIRST 10 ROWS ONLY

Also, run SELECT COUNT(*) to see the number of rows in a table

Step 4: Trace How It's Used

Work out how the data is used:

- Search the source code for table names or SQL statements
- Look in the database for stored procedures or functions

Step 5: Document As You Go

Start a simple document (Confluence, Notion, Markdown file, anything)
Capture:

- what the table stores
- how it connects to others
- any quirks or data issues
- anything that looks wrong or missing

Share this with your team!