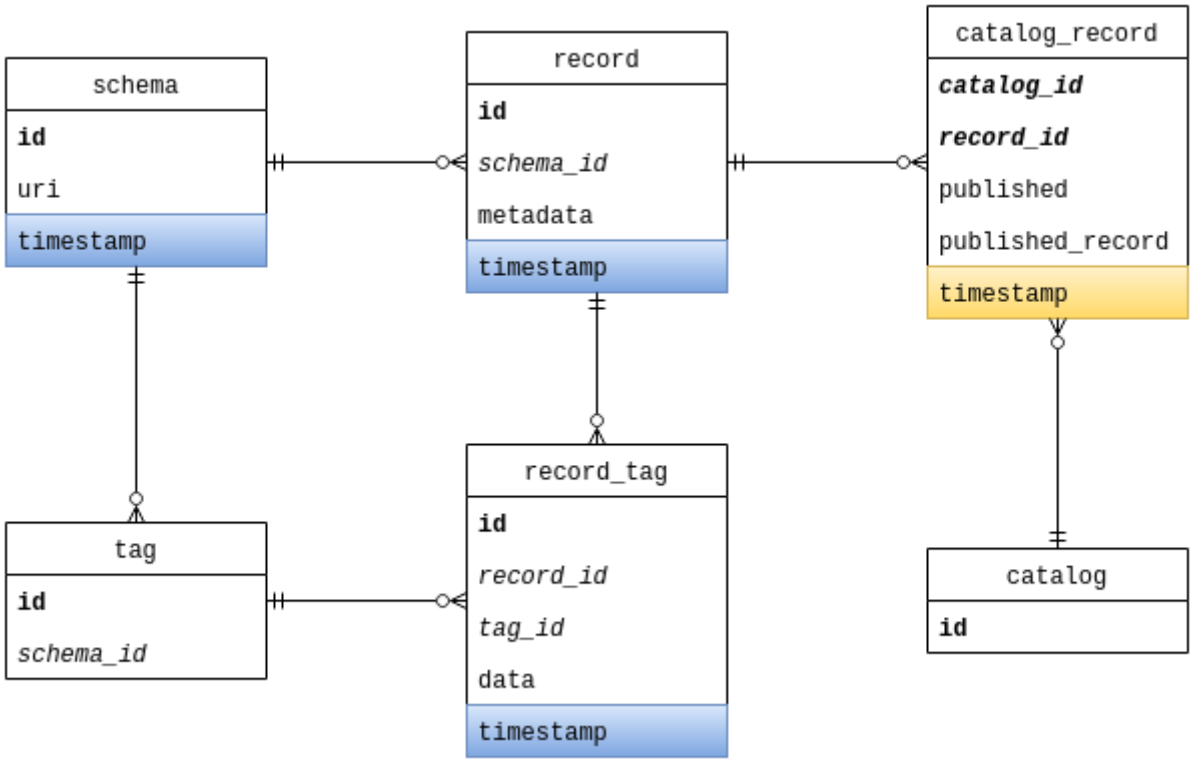
SAEON Developer SQL Test

Introduction

The entity-relationship diagram shown below depicts a part of the data model – in SAEON's Open Data Platform – relevant to the curation and publication of metadata that enable the discovery and citation of scientific datasets.

Each metadata `record` is associated with a `schema` that is used to validate the metadata. Records may optionally be tagged by curators with quality metrics and other information, stored in `record_tag` objects. Each `tag` definition is itself associated with a `schema` that is used to validate any attached tag data.



The `catalog_record` table represents the state of a record with respect to a given public `catalog`. A background process runs at intervals, (re-)evaluating records for publication, and creating or updating `catalog_record` entries. The `timestamp` column (highlighted yellow) contains the time-stamp of the most recent contributing change (originating from one of the highlighted blue timestamps) for the record at the time of evaluation.

Test Setup

A database setup script is provided, which creates the tables shown above along with test data. The script has been tested on PostgreSQL version 14.

Please use **PostgreSQL** to write your queries and generate your output.

The supplied Docker Compose file may be used to start up a PostgreSQL test database. Note that you will need Docker (with the Compose plugin) installed on your system. The instructions below make use of the PostgreSQL client program, *psql*.

Start the containerized test database:

```
docker compose up -d
```

Create the database tables and test data:

```
psql -h localhost -p 7890 test user -f db.sql
```

Open an interactive session:

```
psql -h localhost -p 7890 test user
```

When prompted for a password, type `pass`.

SQL Test

Question 1

Write a query that selects rows consisting of `record_id, timestamp` tuples from the `catalog_record` table, representing the set of previously evaluated records for the catalog with id `'saeon'`.

Question 2

Write a query that selects rows consisting of `record_id, max_timestamp` tuples, indicating the timestamp of the most recent change for every record in the `record` table. This may be the timestamp of the record itself, the timestamp of the associated metadata schema, the timestamp of an attached record tag, or the timestamp of the schema for an attached tag.

Question 3

Using your Q1 and Q2 queries as subqueries, write a query that selects rows consisting of `record_id, max_timestamp` tuples, for records that need to be (re-)evaluated for publication to the catalog with id `'saeon'`, where `max_timestamp` is the most recent contributing timestamp for each such record.

The result set should include a record if and only if:

- it has not yet previously been evaluated with respect to the catalog – that is, there is no corresponding entry in `catalog_record`; or
- any change has been made since the record was previously evaluated that might affect its publication status – that is, the most recent contributing timestamp for the record is newer than the corresponding `catalog_record.timestamp`.

Question 4

Write a query that selects rows consisting of `record_id, tag_id, value` tuples, obtained from the `record_tag` table. If the tag id is `'qc'` (Quality Control), the value is that of `"pass_"` in `record_tag.data`. If the tag id is `'sdg'` (UN Sustainable Development Goal), the value is that of `"sdg"` in `record_tag.data`.

Question 5

Using your Q4 query as input (adapted with type casts as needed), write a query that produces a pivot table consisting of a single row for every record in the `record` table, with associated `record_tag` data values distributed across columns. The result set should consist of `record_id, qc, sdg1, sdg2` tuples, with null values where corresponding tag data does not exist.

Test Submission

Your answer to each question should include the applicable SQL query, along with the output as printed by your SQL client software.

Please provide your SQL queries and outputs in plain text files. The ordering of the result sets and the exact layout and formatting of the printed output are not important.