

Setting up the Database for Process Mining

To run and use **Scheer PAS Process Mining** you need a database to store the collected statistical and tracing data. Process Mining can be connected to a MySQL, an Oracle, or an SQLServer database. To set up the analytic database, you need a valid installation of one of these three.



Please note that for the time being it is not possible to use MySQL 8.x to store the process data. This is due to a conflict between a newly added MySQL keyword and a Process Mining table schema having the same name.



When preparing the database installation, please consider the following:

The Process Mining database contains analytic data for statistical analysis and can reach a considerable size. But, in contrast to databases that store application data, it does not need to be highly available and fulfill strict recovery requirements.

To prevent that the Process Mining database reaches an unbearable size, you should

- use a separate database to store the Process Mining data.
- run this separate database within a less strict recovery mode (e.g. NOARCHIVELOG in Oracle).

This may lead to the fact that a point in time recovery is not possible, but this will not cause problems as lost data can be simply reloaded from the BRIDGE logs.

The analytical database is composed of two parts:

1. The first part contains stored procedures and working tables used during the collection of data.
2. The second part contains the front-end tables which are queried by the Process Mining services to present the data in a user interface.

Setting up a MySQL Database

To use a MySQL database, you need to

- create an empty schema.
- grant the Process Mining user `SELECT` privileges on table `mysql.proc`.

```
GRANT SELECT ON 'mysql'.'proc' TO '<user>'@'<mysql server>';
```



Also consider the following hints

- To use a MySQL 5.5 or 5.6 database, you need to adjust the database settings and set `innodb-large-prefix` to true. Refer to the documentation of [innodb-large-prefix](#) for more information on this.
- Make sure that you use [MyISAM](#) as your storage-engine with MySQL to avoid limitations of the MySQL key length (see [Troubleshooting the Process Mining Installation](#)).
- If you want to install the Process Mining database on a MySQL Database using Amazon Web Services RDS, you may get the following error: You do not have the `SUPER` privilege and binary logging is enabled. To solve this problem, enable `log_bin_trust_function_creators` by following the explanations of the [AWS Guide](#).

All tables and procedures will be created by the **analytics-etl-service** at startup.

Setting up an SQLServer Database

To use an SQLServer database, you only need to create an empty schema. All tables and procedures will be created by the **analytics-etl-service** at startup.

Setting up an Oracle Database

On this Page:

- [Setting up a MySQL Database](#)
- [Setting up an SQLServer Database](#)
- [Setting up an Oracle Database](#)
 - [Access to sys.dbms_crypto](#)

Related Pages:

- [How do I enable functions, procedures, and triggers for my Amazon RDS MySQL instance?](#)

Related Documentation:

- [Troubleshooting the Process Mining Installation](#)

To use an Oracle database, you need to create an empty schema. All tables and procedures will be created by the **analytics-etl-service** at startup.

The Oracle database user need to be granted the following minimum privileges for Process Mining to work:

```
GRANT CONNECT, RESOURCE TO <database user the process mining services will use> CONTAINER=CURRENT;
```

Access to sys.dbms_crypto

For both setup scenarios the database administrator needs to grant access to package **sys.dbms_crypto** to the Process Mining database user:

```
GRANT EXECUTE ON SYS.DBMS_CRYPT0 TO <database user the process mining services will use>;
```