1 Roman Pot positions

- **int saveRomanPotPosition(long startTime, long endTime, string label, double value)**

  **Description**: Saves single roman pot position into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

  **Parameters**:
  - `startTime` - validity interval start time.
  - `endTime` - validity interval end time.
  - `label` - roman pot and type label in text format.
  - `value` - roman pot position value.

- **double loadRomanPotPosition(long timestamp, string label)**

  **Description**: Loads latest version of single roman pot position for given timestamp from the database.

  **Parameters**:
  - `timestamp` - timestamp value.
  - `label` - roman pot and type label in text format.

- **double loadRomanPotPositionByVersion(long timestamp, string label, int version)**

  **Description**: Loads specific version of single roman pot position for given timestamp from the database.

  **Parameters**:
2 General Measurements

2.1 LHC Logging Databases data not connected with TOTEM

- **int** `saveGeneralMeasurement`(`long` startTime, `long` endTime, `string` typeLabel, `double` value)

  **Description:** Saves single general measurement into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

  **Parameters:**
  - `startTime` - validity interval start time.
  - `endTime` - validity interval end time.
  - `typeLabel` - measurement type label.
  - `value` - measurement value.

- **int** `saveGeneralMeasurementVector`(`long` startTime, `long` endTime, `string` typeLabel, `vector<double>` values)

  **Description:** Saves general measurements vector into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

  **Parameters:**
  - `startTime` - validity interval start time.
  - `endTime` - validity interval end time.
  - `typeLabel` - measurement type label.
  - `values` - measurement values vector.

- **double** `loadGeneralMeasurement`(`long` timestamp, `string` typeLabel)

  **Description:** Loads latest version of single general measurement for given timestamp from the database.

  **Parameters:**
  - `timestamp` - timestamp value.
  - `typeLabel` - measurement type label.
• vector<double> loadGeneralMeasurementVector(long timestamp, string typeLabel)

Description: Loads latest version of general measurements vector for given timestamp from the database.

Parameters:
– timestamp - timestamp value.
– typeLabel - measurement type label.

• double loadGeneralMeasurementByVersion(long timestamp, string typeLabel, int version)

Description: Loads latest version of single general measurement for given timestamp from the database.

Parameters:
– timestamp - timestamp value.
– typeLabel - measurement type label.
– version - version number.

• vector<double> loadGeneralMeasurementVectorByVersion(long timestamp, string typeLabel, int version)

Description: Loads latest version of general measurements vector for given timestamp from the database.

Parameters:
– timestamp - timestamp value.
– typeLabel - measurement type label.
– version - version number.

2.2 Currents production

• int saveCurrents(long startTime, long endTime, map<string, double> currents)

Description: Saves currents data into the database. The method returns version number of saved data (in case of multiple similar inserts the old one is not removed, but new version of the same data is created instead).

Parameters:
– startTime - validity interval start time.
– endTime - validity interval end time.
– currents - label - value currents map.
3 Run Information

- **void saveRunInformation**(int runNumber, string description, vector<Event> events)

  **Description**: Saves run information into the database.
  **Parameters**:
  - `runNumber` - run number.
  - `description` - label describing run.
  - `events` - events sequence, the structure is described below.

- **Event** - a structure representing single event.
  **Fields**:
  - `long timestamp` - a timestamp.
  - `int block` - a block number.
  - `int number` - an event number.