Servers

Servers 1

- Local_Development

Copyright © All Rights Reserved

Local_Development

Databases 2

- data_db
- static_db

Users 3

- dataCollector@%
- databaseManager@%
- reportGenerator@%

Properties

Name	Value
Product	MySQL
Version	8.3.0
Version Comment	MySQL Community Server - GPL
Collation	utf8mb4_0900_ai_ci

Copyright © All Rights Reserved

Databases

Databases 2

- data_db
- static_db

Copyright © All Rights Reserved

data_db

Properties

Name	Value
Collation	utf8mb4_0900_ai_ci

Object Types 4

- Tables
- Procedures
- Functions
- Triggers

Copyright © All Rights Reserved

Tables

Objects 3

Name	Description
data	Data Recollected from PLC
sample	Test Sample information
specimen	Test specimen

Copyright © All Rights Reserved

data

Description

Data Recollected from PLC

Properties

Name	Value
Engine	InnoDB
Auto Increment	3887
Average Row Length	58
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:05:27
Last Modified	1/1/0001 00:00:00

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	specimen	INT		11		True	False	False	True	False		False	False	
	pressure	DOUBLE	0			False	False	False	True	False		False	False	
	temperature	DOUBLE	0			False	False	False	True	False		False	False	
	createdAt	DATETIME		4		False	False	False	False	False	CURRENT_TIMESTAMP(4)	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	IDX_test_data_test_specimen	specimen	False	None	0

Foreign Keys

Name	Columns	Delete Rule	Update Rule

Copyright © All Rights Reserved

FK_test_data_test_specimen_id	id	N/S	N/S

SQL Script

```
CREATE TABLE data (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  specimen int UNSIGNED NOT NULL,
  pressure double NOT NULL,
  temperature double NOT NULL,
  createdAt datetime(4) DEFAULT CURRENT_TIMESTAMP(4),
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 58,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Data Recollected from PLC';

ALTER TABLE data
ADD INDEX IDX_test_data_test_specimen (specimen);

ALTER TABLE data
ADD CONSTRAINT FK_test_data_test_specimen_id FOREIGN KEY (specimen)
REFERENCES specimen (id);
```

Depends On 1

- specimen

Used By 6

- insertData
- DeleteTest
- selectTestData
- getSpecimenData
- selectTestData
- selectTestLimit

Copyright © All Rights Reserved

sample

Description

Test Sample information

Properties

Name	Value
Engine	InnoDB
Auto Increment	50
Average Row Length	16384
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:05:27
Last Modified	1/1/0001 00:00:00

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	standard	VARCHAR	25			False	False	False	True	False		False	False	
	material	VARCHAR	10			False	False	False	True	False		False	False	
	specification	VARCHAR	10			False	False	False	True	False		False	False	
	diamreal	INT		11		True	False	False	True	False		False	False	
	diamnom	INT		11		True	False	False	True	False		False	False	
	wallthick	INT		11		True	False	False	True	False		False	False	
	lentotal	INT		11		True	False	False	True	False		False	False	
	lenfree	INT		11		True	False	False	True	False		False	False	
	targettemp	INT		11		False	False	False	True	False		False	False	
	targetpressure	INT		11		False	False	False	True	False		False	False	
	condPeriod	VARCHAR	15			False	False	False	True	False		False	False	

Copyright © All Rights Reserved

	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0

Triggers

Name	Type	Event	Security	Definer
NewSample	BEFORE	INSERT	Definer	root

SQL Script

```
CREATE TABLE sample (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  standard varchar(25) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  material varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  specification varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  diamreal int UNSIGNED NOT NULL,
  diamnom int UNSIGNED NOT NULL,
  wallthick int UNSIGNED NOT NULL,
  lentotal int UNSIGNED NOT NULL,
  lenfree int UNSIGNED NOT NULL,
  targettemp int NOT NULL,
  targetpressure int NOT NULL,
  condPeriod varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 16384,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Test Sample information';
```

Depends On

No items found

Used By 12

- insertSample
- insertSpecimen
- selectSample
- selectSampleID
- selectCompareTests
- selectTest
- selectTests
- NewSample
- specimen
- getSpecimenData
- selectTestSample
- DeleteTestSample

Copyright © All Rights Reserved

specimen

Description

Test specimen

Properties

Name	Value
Engine	InnoDB
Auto Increment	136
Average Row Length	910
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:05:27
Last Modified	1/1/0001 00:00:00

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	sample	INT		11		True	False	False	True	False		False	False	
	operator	VARCHAR	40			False	False	False	True	False	'STEL S.A.'	False	False	
	enviroment	VARCHAR	20			False	False	False	False	False	''	False	False	
	testName	VARCHAR	120			False	False	False	False	False	''	False	False	
	endCap	VARCHAR	10			False	False	False	False	False	''	False	False	
	failText	VARCHAR	200			False	False	False	True	False	'Sin Fallas'	False	False	
	remark	VARCHAR	200			False	False	False	True	False	'Sin Observaciones'	False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Copyright © All Rights Reserved

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_test_specimen_test_sample_id	sample	False	None	0

Triggers

Name	Type	Event	Security	Definer
DeleteTest	BEFORE	DELETE	Definer	root
DeleteTestSample	AFTER	DELETE	Definer	root

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_test_specimen_test_sample_id	id	N/S	N/S

SQL Script

```sql
CREATE TABLE specimen (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  sample int UNSIGNED NOT NULL,
  operator varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'STEL S.A.',
  enviroment varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT '',
  testName varchar(120) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT '',
  endCap varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT '',
  failText varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'Sin Fallas',
  remark varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT 'Sin Observaciones',
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 910,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Test specimen';

ALTER TABLE specimen
ADD CONSTRAINT FK_test_specimen_test_sample_id FOREIGN KEY (sample)
REFERENCES sample (id);
```

Depends On 1

- sample

Used By 17

- countSpecimens
- insertData
- insertSpecimen
- selectSpecimen
- selectCompareTests
- selectTest
- updateSpecimen
- data
- DeleteTest
- DeleteTestSample
- deleteTest
- getSpecimenData
- countSpecimens
- selectTestNumber
- selectTestSpecimen
- selectTestsSpecimens
- specimenExists

Copyright © All Rights Reserved

Procedures

Objects 14

Name	Description
countSpecimens	
deleteTest	
getSpecimenData	
insertData	
insertSample	
insertSpecimen	
selectCompareTests	
selectSample	
selectSampleID	
selectSpecimen	
selectTest	
selectTestData	
selectTests	
updateSpecimen	

Copyright © All Rights Reserved

countSpecimens

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE countSpecimens (IN idSample int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    COUNT(ts.sample) AS counts
  FROM specimen ts
  WHERE ts.sample = idSample;

END
```

Depends On 2

- dataCollector@%
- specimen

Used By 1

- selectTestSpecimen

Copyright © All Rights Reserved

deleteTest

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteTest (IN idSpecimen int UNSIGNED)
BEGIN

  DELETE LOW_PRIORITY QUICK
    FROM specimen ts
  WHERE ts.id = idSpecimen;

END
```

Depends On 1

specimen

Used By

No items found

Copyright © All Rights Reserved

getSpecimenData

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
mySpecimenID	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE getSpecimenData (IN mySpecimenID int UNSIGNED)
DETERMINISTIC
BEGIN

  CREATE TEMPORARY TABLE IF NOT EXISTS myResult AS (SELECT
      td.id,
      td.pressure,
      td.temperature,
      # td.createdAt,
      (TIMESTAMPDIFF(MICROSECOND, (SELECT
          td2.createdAt
        FROM data td2
        WHERE td2.specimen = mySpecimenID LIMIT 1), td.createdAt)) AS `difference`
    FROM data td
    WHERE td.specimen = mySpecimenID);

  SELECT
    ts.id AS `idTest`,
    (SELECT
        JSON_ARRAYAGG(JSON_OBJECT('id', r.id, 'pressure', r.pressure, 'temperature', r.temperature, 'timekey', r.differe
nce))
      FROM myResult r) AS `testData`
  FROM specimen ts
    INNER JOIN sample ts1
      ON ts.sample = ts1.id
  WHERE ts.id = mySpecimenID;

  DROP TEMPORARY TABLE IF EXISTS myResult;

END
```

Depends On 3

Copyright © All Rights Reserved

data

specimen

sample

Used By

No items found

Copyright © All Rights Reserved

insertData

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
pressure	DOUBLE	0		
temperature	DOUBLE	0		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertData (IN idSpecimen int UNSIGNED, IN pressure double, IN temperature double)
BEGIN

  INSERT INTO data (specimen, pressure, temperature, createdAt)
    VALUES (idSpecimen, pressure, temperature, NOW());

END
```

Depends On 3

- dataCollector@%
- data
- specimen

Used By

No items found

Copyright © All Rights Reserved

insertSample

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
standard	VARCHAR	25		
material	VARCHAR	10		
specification	VARCHAR	6		
diamnom	INT		11	
diamreal	INT		11	
wallthick	INT		11	
lenfree	INT		11	
lentotal	INT		11	
targetpressure	INT		11	
targettemp	INT		11	
condPeriod	VARCHAR	15		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertSample (IN standard varchar(25), IN material varchar(10), IN specification varchar(6), IN diamnom
int UNSIGNED, IN diamreal int UNSIGNED, IN wallthick int UNSIGNED, IN lenfree int UNSIGNED, IN lentotal int
UNSIGNED, IN targetpressure int, IN targettemp int, IN condPeriod varchar(15))
DETERMINISTIC
BEGIN

  DECLARE EXIT HANDLER FOR SQLEXCEPTION
  BEGIN

    GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE, @message = MESSAGE_TEXT;
    SELECT
      @message AS `idSample`;

  END;
```


Copyright © All Rights Reserved

```
  INSERT INTO sample (standard, condPeriod, material, specification, targettemp, targetpressure, diamreal, diamnom,
  wallthick, lenfree, lentotal)
    VALUES (standard, condPeriod, material, specification, targettemp, targetpressure, diamreal, diamnom, wallthick
    , lenfree, lentotal);

  SELECT
    ts.id AS `idSample`
  FROM sample ts
  ORDER BY ts.id DESC
  LIMIT 1;

END
```

Depends On 2

- dataCollector@%
- sample

Used By

No items found

Copyright © All Rights Reserved

insertSpecimen

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	
operatorName	VARCHAR	40		
enviromentIn	VARCHAR	20		
testNameIn	VARCHAR	120		
endCapIn	VARCHAR	10		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertSpecimen (IN idSample int UNSIGNED, IN operatorName varchar(40), IN enviromentIn varchar(20), IN
testNameIn varchar(120), IN endCapIn varchar(10))
BEGIN

  INSERT INTO specimen (sample, operator, enviroment, testName, endCap, createdAt, updatedAt)
    VALUES (idSample, operatorName, enviromentIn, testNameIn, endCapIn, CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP());

  SELECT
    ts.id AS `idSpecimen`
  FROM specimen ts
  ORDER BY ts.id DESC
  LIMIT 1;

END
```

Depends On 3

- dataCollector@%
- specimen
- sample

Used By

No items found

Copyright © All Rights Reserved

Copyright © All Rights Reserved

selectCompareTests

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
arraySpecimens	TEXT			

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectCompareTests (IN arraySpecimens text)
DETERMINISTIC
BEGIN

  SELECT
    (selectTestSample(sa.id)) AS 'mySample',
    (selectTestSpecimen(se.id)) AS 'mySpecimen',
    (selectTestData(se.id)) AS 'myData'
  FROM specimen se
    INNER JOIN sample sa
      ON se.sample = sa.id
  WHERE FIND_IN_SET(se.id, arraySpecimens) > 0;

END
```

Depends On 6

- reportGenerator@%
- selectTestSample
- selectTestSpecimen
- selectTestData
- specimen
- sample

Used By

No items found

Copyright © All Rights Reserved

selectSample

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectSample (IN idSample int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    s.id AS `idSample`,
    s.standard AS `standard`,
    s.material AS `material`,
    s.specification AS `specification`,
    s.diamreal AS `diamreal`,
    s.diamnom AS `diamnom`,
    s.wallthick AS `wallthickness`,
    s.lenfree AS `lenfree`,
    s.lentotal AS `lentotal`,
    s.condPeriod AS `condPeriod`,
    s.targettemp AS `targettemp`,
    s.targetpressure AS `targetpressure`
  FROM sample s
  WHERE s.id = idSample;

END
```

Depends On 2

- dataCollector@%
- sample

Used By

No items found

Copyright © All Rights Reserved

selectSampleID

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
standard	VARCHAR	25		
material	VARCHAR	10		
specification	VARCHAR	6		
diamnom	INT		11	
diamreal	INT		11	
wallthick	INT		11	
lenfree	INT		11	
lentotal	INT		11	
targetpressure	INT		11	
targettemp	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectSampleID (IN standard varchar(25), IN material varchar(10), IN specification varchar(6), IN diamnom
int UNSIGNED, IN diamreal int UNSIGNED, IN wallthick int UNSIGNED, IN lenfree int UNSIGNED, IN lentotal int
UNSIGNED, IN targetpressure int, IN targettemp int)
DETERMINISTIC
BEGIN

  SELECT
    ts.id AS idSample
  FROM sample ts
  WHERE (ts.standard LIKE standard
  AND ts.material LIKE material
  AND ts.specification LIKE specification
  AND ts.diamreal = diamreal
  AND ts.diamnom = diamnom
  AND ts.wallthick = wallthick
  AND ts.lenfree = lenfree
```

 Copyright © All Rights Reserved

```
    AND ts.lentotal = lentotal
    AND ts.targettemp = targettemp
    AND ts.targetpressure = targetpressure);

END
```

Depends On 2

- dataCollector@%
- sample

Used By

No items found

Copyright © All Rights Reserved

selectSpecimen

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectSpecimen (IN idSpecimen int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    s.id AS `idSpecimen`,
    s.sample AS `idSample`,
    s.enviroment AS `enviroment`,
    s.testName AS `testName`,
    s.operator AS `operator`,
    s.endCap AS `endCap`
  FROM specimen s
  WHERE s.id = idSpecimen;

END
```

Depends On 2

- dataCollector@%
- specimen

Used By

No items found

Copyright © All Rights Reserved

selectTest

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectTest (IN idSpecimen int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    (selectTestSample(sa.id)) AS 'mySample',
    (selectTestSpecimen(se.id)) AS 'mySpecimen'
  FROM specimen se
    INNER JOIN sample sa
      ON se.sample = sa.id
  WHERE se.id = idSpecimen;

END
```

Depends On 5

- reportGenerator@%
- selectTestSample
- selectTestSpecimen
- specimen
- sample

Used By

No items found

Copyright © All Rights Reserved

selectTestData

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectTestData (IN idSpecimen int UNSIGNED)
DETERMINISTIC
BEGIN

  SET @initTime = (SELECT
      td.createdAt
    FROM data td
    WHERE td.specimen = idSpecimen
    ORDER BY td.createdAt ASC LIMIT 1);

  WITH groupingData
  AS
  (SELECT
      td.createdAt,
      AVG(td.pressure) AS 'pressure',
      AVG(td.temperature) AS 'temperature'
    FROM data td
    WHERE td.specimen = idSpecimen
    GROUP BY td.createdAt)

  SELECT
    TIMESTAMPDIFF(SECOND, @initTime, gd.createdAt) AS 'key',
    gd.pressure AS 'pressure',
    gd.temperature AS 'temperature'
  FROM groupingData gd;

END
```

Depends On 2

- reportGenerator@%
- data

Copyright © All Rights Reserved

Used By 1

selectCompareTests

Copyright © All Rights Reserved

selectTests

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectTests ()
DETERMINISTIC
BEGIN

  SELECT
    ts.id AS idSample,
    ts.standard AS standard,
    ts.material AS material,
    selectTestsSpecimens(ts.id) AS mySpecimens
  FROM sample ts;

END
```

Depends On 3

- reportGenerator@%
- selectTestsSpecimens
- sample

Used By

No items found

Copyright © All Rights Reserved

updateSpecimen

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
testName	VARCHAR	150		
operator	VARCHAR	40		
fail	VARCHAR	255		
remark	VARCHAR	255		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateSpecimen (IN idSpecimen int UNSIGNED, IN testName varchar(150), IN operator varchar(40), IN fail
varchar(255), IN remark varchar(255))
BEGIN

  UPDATE LOW_PRIORITY specimen ts
  SET ts.testName = testName,
      ts.operator = operator,
      ts.failText = fail,
      ts.remark = remark,
      ts.updatedAt = CURRENT_TIMESTAMP()
  WHERE ts.id = idSpecimen;

END
```

Depends On 2

- reportGenerator@%
- specimen

Used By

No items found

Copyright © All Rights Reserved

Functions

Objects 8

Name	Description
countSpecimens	
selectTestData	
selectTestLimit	
selectTestNumber	
selectTestSample	
selectTestSpecimen	
selectTestsSpecimens	
specimenExists	

Copyright © All Rights Reserved

countSpecimens

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
(Result)	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION countSpecimens (idSpecimen int UNSIGNED)
RETURNS int(11) UNSIGNED
DETERMINISTIC
BEGIN

  SET @count = 0;

  SELECT
    COUNT(*) INTO @count
  FROM specimen ts
  WHERE ts.sample = (SELECT
      se.sample
    FROM specimen se
    WHERE se.id = idSpecimen)
  GROUP BY ts.sample;

  RETURN @count;
END
```

Depends On 1

specimen

Used By

No items found

Copyright © All Rights Reserved

selectTestData

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestData (idSpecimen int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  SET @returnData = NULL;

  SET @initTime = (SELECT
      td.createdAt
    FROM data td
    WHERE td.specimen = idSpecimen
    ORDER BY td.createdAt ASC LIMIT 1);

  WITH groupingData
  AS
  (SELECT
      td.createdAt,
      AVG(td.pressure) AS 'pressure',
      AVG(td.temperature) AS 'temperature'
    FROM data td
    WHERE td.specimen = idSpecimen
    GROUP BY td.createdAt)

  SELECT
    JSON_ARRAYAGG(JSON_OBJECT('key',
    TIMESTAMPDIFF(SECOND, @initTime, gd.createdAt),
    'pressure',
    gd.pressure,
    'temperature',
    gd.temperature)) INTO @returnData
  FROM groupingData gd;

  RETURN @returnData;
```


Copyright © All Rights Reserved

```
END
```

Depends On 1

data

Used By

No items found

Copyright © All Rights Reserved

selectTestLimit

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
isASC	TINYINT		4	
(Result)	DATETIME		0	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestLimit (idSpecimen int UNSIGNED, isASC tinyint UNSIGNED)
RETURNS datetime
DETERMINISTIC
BEGIN

  SET @Time = NULL;

  IF isASC = 0 THEN
    SELECT
      td.createdAt INTO @Time
    FROM data td
    WHERE td.specimen = idSpecimen
    ORDER BY td.createdAt ASC LIMIT 1;
  ELSE
    SELECT
      td.createdAt INTO @Time
    FROM data td
    WHERE td.specimen = idSpecimen
    ORDER BY td.createdAt DESC LIMIT 1;
  END IF;

  RETURN @Time;
END
```

Depends On 1

data

Copyright © All Rights Reserved

Used By 2

- selectTestSpecimen
- selectTestsSpecimens

Copyright © All Rights Reserved

selectTestNumber

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
(Result)	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestNumber (idSpecimen int UNSIGNED)
RETURNS int(11) UNSIGNED
DETERMINISTIC
BEGIN
  SET @testNumber = 0;

  SET @sampleID = (SELECT
      ts.sample
    FROM specimen ts
    WHERE ts.id = idSpecimen);

  SELECT
    COUNT(*) INTO @testNumber
  FROM specimen ts
  WHERE ts.id <= idSpecimen
  AND ts.sample = @sampleID
  GROUP BY ts.sample;

  RETURN @testNumber;
END
```

Depends On 1

- specimen

Used By 2

- selectTestSpecimen
- selectTestsSpecimens

Copyright © All Rights Reserved

Copyright © All Rights Reserved

selectTestSample

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestSample (idSample int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  SET @result = NULL;

  SELECT
    JSON_OBJECT('idSample',
    sa.id,
    'standard',
    sa.standard,
    'material',
    sa.material,
    'specification',
    sa.specification,
    'diameterReal',
    sa.diamreal,
    'diameterNominal',
    sa.diamnom,
    'wallThickness',
    sa.wallthick,
    'lengthTotal',
    sa.lentotal,
    'lengthFree',
    sa.lenfree,
    'targetTemperature',
    sa.targettemp,
    'targetPressure',
    sa.targetpressure,
    'hoopStress',
    ((sa.targettemp * (sa.diamreal - sa.wallthick)) / (10 * 2 * sa.wallthick)),
    'conditionalPeriod',
```


Copyright © All Rights Reserved

```
    sa.condPeriod) INTO @result
  FROM sample sa
  WHERE sa.id = idSample;

  RETURN @result;
END
```

Depends On 1

- sample

Used By 2

- selectCompareTests
- selectTest

Copyright © All Rights Reserved

selectTestSpecimen

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSpecimen	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestSpecimen (idSpecimen int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  SET @result = NULL;

  SET @countSpecimens = countSpecimens(idSpecimen);

  SET @beginTime = selectTestLimit(idSpecimen, 0);

  SET @endTime = selectTestLimit(idSpecimen, 1);

  SET @testNumber = selectTestNumber(idSpecimen);

  SELECT
    JSON_OBJECT('idSpecimen',
    se.id,
    'operator',
    se.operator,
    'enviroment',
    se.enviroment,
    'beginTime',
    DATE_FORMAT(@beginTime,
    '%d/%m/%Y %H:%i:%s'),
    'endTime',
    DATE_FORMAT(@endTime,
    '%d/%m/%Y %H:%i:%s'),
    'duration',
    DATE_FORMAT(TIMEDIFF(@endTime,
    @beginTime),
    '%H:%i:%s'),
    'counts',
```


Copyright © All Rights Reserved

```
    @countSpecimens,
    'testNumber',
    @testNumber,
    'testName',
    se.testName,
    'endCap',
    se.endCap,
    'fail',
    se.failText,
    'remark',
    se.remark) INTO @result
  FROM specimen se
  WHERE se.id = idSpecimen;

  RETURN @result;
END
```

Depends On 4

- countSpecimens
- selectTestLimit
- selectTestNumber
- specimen

Used By 2

- selectCompareTests
- selectTest

Copyright © All Rights Reserved

selectTestsSpecimens

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTestsSpecimens (idSample int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN
  SET @myResult = NULL;

  WITH preResults
  AS
  (SELECT
      ts.id,
      ts.operator,
      selectTestNumber(ts.id) AS testNumber,
      (selectTestLimit(ts.id, 0)) AS beginTime,
      (selectTestLimit(ts.id, 1)) AS endTime
    FROM specimen ts
    WHERE ts.sample = idSample)

  (SELECT
      JSON_ARRAYAGG(JSON_OBJECT('idSpecimen', pr.id,
      'operator', pr.operator, 'testNumber', pr.testNumber, 'beginTime', DATE_FORMAT(pr.beginTime, '%d/%m/%Y
%H:%i:%s'), 'endTime', DATE_FORMAT(pr.endTime, '%d/%m/%Y %H:%
      i:%s'), 'duration', DATE_FORMAT(
      TIMEDIFF(pr.endTime, pr.beginTime), '%H:%i:%s'))) INTO @myResult
    FROM preResults pr
    ORDER BY pr.id DESC);

  RETURN @myResult;
END
```

Depends On 3

Copyright © All Rights Reserved

selectTestNumber

selectTestLimit

specimen

Used By 1

selectTests

Copyright © All Rights Reserved

specimenExists

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:05:27
Last Modified	11/6/2024 11:05:27

Parameters

Name	Data Type	Length	Precision	Scale
idSample	INT		11	
start	VARBINARY	255		
(Result)	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION specimenExists (idSample int UNSIGNED, start varbinary(255))
RETURNS int(11) UNSIGNED
DETERMINISTIC
BEGIN

  DECLARE idSpecimen boolean;

  SELECT
    ts.id INTO idSpecimen
  FROM specimen ts
  WHERE (ts.sample = idSample
  AND ts.start = start)
  LIMIT 1;

  RETURN idSpecimen;
END
```

Depends On 1

specimen

Used By

No items found

Copyright © All Rights Reserved

Triggers

Objects 3

Name
DeleteTest
DeleteTestSample
NewSample

 Copyright © All Rights Reserved

DeleteTest

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	DELETE
Timing	BEFORE
Follows	
Precedes	
Created	11/6/2024 11:05:48

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER DeleteTest
BEFORE DELETE
ON specimen
FOR EACH ROW
BEGIN

  DELETE LOW_PRIORITY QUICK
    FROM data td
  WHERE td.specimen = OLD.id;

END
```

Depends On 3

- reportGenerator@%
- data
- specimen

Used By

No items found

Copyright © All Rights Reserved

DeleteTestSample

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	DELETE
Timing	AFTER
Follows	
Precedes	
Created	11/6/2024 11:05:48

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER DeleteTestSample
AFTER DELETE
ON specimen
FOR EACH ROW
BEGIN

  SET @countTests = 0;

  SELECT
    COUNT(*) INTO @countTests
  FROM specimen ts
  WHERE ts.sample = OLD.sample;

  IF @countTests < 1 THEN

    DELETE LOW_PRIORITY QUICK
      FROM sample
    WHERE sample.id = OLD.sample;

  END IF;

END
```

Depends On 2

- specimen
- sample

Used By

No items found

Copyright © All Rights Reserved

NewSample

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	INSERT
Timing	BEFORE
Follows	
Precedes	
Created	11/6/2024 11:05:48

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER NewSample
BEFORE INSERT
ON sample
FOR EACH ROW
BEGIN

  DECLARE row_id int UNSIGNED;

  SELECT
    s.id INTO row_id
  FROM sample s
  WHERE s.standard LIKE NEW.standard
  AND s.material LIKE NEW.material
  AND s.specification LIKE NEW.specification
  AND s.diamreal = NEW.diamreal
  AND s.diamnom = NEW.diamnom
  AND s.lentotal = NEW.lentotal
  AND s.lenfree = NEW.lenfree
  AND s.wallthick = NEW.wallthick
  AND s.targettemp = new.targettemp
  AND s.targetpressure = NEW.targetpressure;

  IF row_id IS NOT NULL THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = row_id;
  END IF;

END
```

Depends On 1

sample

Used By

No items found

Copyright © All Rights Reserved

static_db

Properties

Name	Value
Collation	utf8mb4_0900_ai_ci

Object Types 4

- Tables
- Procedures
- Functions
- Triggers

Copyright © All Rights Reserved

Tables

Objects 10

Name	Description
conditional_period	Conditional Period related to Standard
endcap	
enviroment	Enviroments
material	Material of Hoop
material_has_standard	
operator	
specification	Specification of material
specification_configuration	Conditional period by Specification
standard	Standards: ISO, IRAM, ...
test_type	

Copyright © All Rights Reserved

conditional_period

Description

Conditional Period related to Standard

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	3276
Charset	utf8mb4
Collation	utf8mb4_unicode_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	24/6/2024 22:36:05
Last Modified	24/6/2024 23:37:02

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	standard	INT		11		True	False	False	True	False		False	False	
	time	INT		11		True	False	False	True	False	'0'	False	False	
	timeType	VARCHAR	3			False	False	False	True	False	'h'	False	False	
	aproxTime	INT		11		True	False	False	True	False	'0'	False	False	
	aproxType	VARCHAR	3			False	False	False	True	False	'min'	False	False	
	minwall	INT		11		True	False	False	True	False		False	False	
	maxwall	INT		11		True	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedtAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Copyright © All Rights Reserved

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_conditional_period_standard_id	standard	False	None	0

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_conditional_period_standard_id	id	N/S	N/S

SQL Script

```sql
CREATE TABLE conditional_period (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  standard int UNSIGNED NOT NULL,
  time int UNSIGNED NOT NULL DEFAULT 0,
  timeType varchar(3) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT 'h',
  aproxTime int UNSIGNED NOT NULL DEFAULT 0,
  aproxType varchar(3) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT 'min',
  minwall int UNSIGNED NOT NULL,
  maxwall int UNSIGNED NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedtAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 3276,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_unicode_ci,
COMMENT = 'Conditional Period related to Standard';

ALTER TABLE conditional_period
ADD CONSTRAINT FK_conditional_period_standard_id FOREIGN KEY (standard)
REFERENCES standard (id);
```

Depends On 1

- standard

Used By 7

- selectConditionalPeriods
- deleteConditionalPeriod
- insertConditionalPeriod
- selectStandarsComplete
- existConditionalPeriod
- selectConditionalPeriods_has_Standard
- deleteStandardChilds

Copyright © All Rights Reserved

endcap

Description

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	5461
Charset	latin1
Collation	latin1_swedish_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	24/6/2024 18:35:57

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	standard	INT		11		True	False	False	True	False		False	False	
	name	VARCHAR	15			False	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_endcap_standard_id	standard	False	None	0

Foreign Keys

Copyright © All Rights Reserved

Name	Columns	Delete Rule	Update Rule
FK_endcap_standard_id	id	N/S	N/S

SQL Script

```
CREATE TABLE endcap (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  standard int UNSIGNED NOT NULL,
  name varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 5461,
CHARACTER SET latin1,
COLLATE latin1_swedish_ci;

ALTER TABLE endcap
ADD CONSTRAINT FK_endcap_standard_id FOREIGN KEY (standard)
REFERENCES standard (id) ON DELETE CASCADE ON UPDATE CASCADE;
```

Depends On 1

- standard

Used By 7

- selectEndCaps
- deleteEndCap
- insertEndCap
- selectStandarsComplete
- existEndCap
- selectEndCaps_has_Standard
- deleteStandardChilds

Copyright © All Rights Reserved

enviroment

Description

Enviroments

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	4096
Charset	latin1
Collation	latin1_swedish_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	24/6/2024 18:52:43

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	standard	INT		11		True	False	False	True	False		False	False	
	inside	VARCHAR	30			False	False	False	True	False		False	False	
	outside	VARCHAR	30			False	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_enviroment_standard_id	standard	False	None	0

Copyright © All Rights Reserved

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_enviroment_standard_id	id	N/S	N/S

SQL Script

```
CREATE TABLE enviroment (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  standard int UNSIGNED NOT NULL,
  inside varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  outside varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 4096,
CHARACTER SET latin1,
COLLATE latin1_swedish_ci,
COMMENT = 'Enviroments';

ALTER TABLE enviroment
ADD CONSTRAINT FK_enviroment_standard_id FOREIGN KEY (standard)
REFERENCES standard (id) ON DELETE CASCADE ON UPDATE CASCADE;
```

Depends On 1

- standard

Used By 7

- selectEnviroment
- deleteEnviroment
- insertEnviroment
- selectStandarsComplete
- existEnviroment
- selectEnviroments_has_Standard
- deleteStandardChilds

Copyright © All Rights Reserved

material

Description

Material of Hoop

Properties

Name	Value
Engine	InnoDB
Auto Increment	14
Average Row Length	16384
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	25/6/2024 00:48:45
Last Modified	24/6/2024 16:36:35

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	name	VARCHAR	15			False	False	False	True	False		False	False	
	description	VARCHAR	120			False	False	False	True	False	'Sin descripción'	False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0

Triggers

Name	Type	Event	Security	Definer

Copyright © All Rights Reserved

deleteMaterialChilds	BEFORE	DELETE	Definer	root

Permissions

Action	Owner
SELECT	databaseManager@%

SQL Script

```
CREATE TABLE material (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  name varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  description varchar(120) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT 'Sin descripción',
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 16384,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Material of Hoop';
```

Depends On 1

- databaseManager@%

Used By 13

- selectMaterials
- deleteMaterial
- insertMaterial
- insertRelatedMaterial
- insertSpecification
- selectMaterialsJSON
- updateMaterial
- deleteMaterialChilds
- material_has_standard
- specification
- selectStandarsComplete
- existMaterial
- selectMaterials_has_Standard

Copyright © All Rights Reserved

material_has_standard

Description

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	8192
Charset	utf8mb4
Collation	utf8mb4_0900_ai_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	24/6/2024 16:36:44

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	material	INT		11		True	False	False	True	False		False	False	
	standard	INT		11		True	False	False	True	False		False	False	
	type	VARCHAR	25			False	False	False	True	False	'Plastico'	False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_material_has_standard_material_id	material	False	None	0

Copyright © All Rights Reserved

	FK_material_has_standard_standard_id	standard	False	None	0

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_material_has_standard_material_id	id	N/S	N/S
FK_material_has_standard_standard_id	id	N/S	N/S

SQL Script

```
CREATE TABLE material_has_standard (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  material int UNSIGNED NOT NULL,
  standard int UNSIGNED NOT NULL,
  type varchar(25) NOT NULL DEFAULT 'Plastico',
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 8192,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

ALTER TABLE material_has_standard
ADD CONSTRAINT FK_material_has_standard_material_id FOREIGN KEY (material)
REFERENCES material (id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE material_has_standard
ADD CONSTRAINT FK_material_has_standard_standard_id FOREIGN KEY (standard)
REFERENCES standard (id) ON DELETE CASCADE ON UPDATE CASCADE;
```

Depends On 2

- material
- standard

Used By 8

- selectMaterials
- deleteRelatedMaterial
- insertRelatedMaterial
- selectStandarsComplete
- existRelatedMaterial
- selectMaterials_has_Standard
- deleteMaterialChilds
- deleteStandardChilds

Copyright © All Rights Reserved

operator

Description

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	2730
Charset	utf8mb4
Collation	utf8mb4_0900_ai_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:03
Last Modified	1/1/0001 00:00:00

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	dni	INT		11		True	False	False	True	False		False	False	
	name	VARCHAR	50			False	False	False	True	False		False	False	
	familyname	VARCHAR	50			False	False	False	True	False		False	False	
	createdAt	DATETIME		4		False	False	False	True	False	CURRENT_TIMESTAMP(4)	False	False	
	updatedAt	DATETIME		4		False	False	False	True	False	CURRENT_TIMESTAMP(4)	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0

SQL Script

```
CREATE TABLE operator (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  dni int UNSIGNED NOT NULL,
```


Copyright © All Rights Reserved

```
  name varchar(50) NOT NULL,
  familyname varchar(50) NOT NULL,
  createdAt datetime(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4),
  updatedAt datetime(4) NOT NULL DEFAULT CURRENT_TIMESTAMP(4) ON UPDATE CURRENT_TIMESTAMP(4),
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 2730,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;
```

Depends On

No items found

Used By 6

- selectOperators
- deleteOperator
- insertOperator
- selectOperatorsJSON
- updateOperator
- existOperator

Copyright © All Rights Reserved

specification

Description

Specification of material

Properties

Name	Value
Engine	InnoDB
Auto Increment	22
Average Row Length	16384
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	25/6/2024 00:48:11
Last Modified	24/6/2024 23:46:56

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	material	INT		11		True	False	False	True	False		False	False	
	name	VARCHAR	15			False	False	False	True	False		False	False	
	description	VARCHAR	120			False	False	False	True	False	'Sin Descripción'	False	False	
	createdAt	DATETIME		0		False	False	False	True	False		False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_specification_material_id	material	False	None	0

Copyright © All Rights Reserved

Triggers

Name	Type	Event	Security	Definer
deleteConfigurationsChilds	BEFORE	DELETE	Definer	root

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_specification_material_id	id	N/S	N/S

SQL Script

```sql
CREATE TABLE specification (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  material int UNSIGNED NOT NULL,
  name varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  description varchar(120) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT 'Sin Descripción',
  createdAt datetime NOT NULL,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 16384,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Specification of material';

ALTER TABLE specification
ADD CONSTRAINT FK_specification_material_id FOREIGN KEY (material)
REFERENCES material (id) ON UPDATE CASCADE;
```

Depends On 1

- material

Used By 12

- selectSpecifications
- deleteSpecification
- insertSpecification
- insertSpecification_Configuration
- updateSpecification
- deleteConfigurationsChilds
- specification_configuration
- existConfiguration
- existSpecification
- existSpecimen
- selectSpecifications
- deleteMaterialChilds

Copyright © All Rights Reserved

specification_configuration

Description

Conditional period by Specification

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	8192
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	1/1/0001 00:00:00

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	specification	INT		11		True	False	False	True	False		False	False	
	time	SMALLINT		6		True	False	False	True	False		False	False	
	timeType	VARCHAR	1			False	False	False	True	False	'H'	False	False	
	temperature	TINYINT		4		True	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0

Copyright © All Rights Reserved

	FK_period_by_material_material_id	specification	False	None	0

Foreign Keys

Name	Columns	Delete Rule	Update Rule
FK_specification_configuration_specification_id	id	N/S	N/S

SQL Script

```sql
CREATE TABLE specification_configuration (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  specification int UNSIGNED NOT NULL,
  time smallint UNSIGNED NOT NULL,
  timeType varchar(1) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL DEFAULT 'H',
  temperature tinyint UNSIGNED NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 8192,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Conditional period by Specification';

ALTER TABLE specification_configuration
ADD INDEX FK_period_by_material_material_id (specification);

ALTER TABLE specification_configuration
ADD CONSTRAINT FK_specification_configuration_specification_id FOREIGN KEY (specification)
REFERENCES specification (id) ON UPDATE CASCADE;
```

Depends On 1

- specification

Used By 7

- selectSettingsSpecification
- deleteSpecification_Configuration
- insertSpecification_Configuration
- updateSpecification_Configuration
- existSpecification_Configuration
- selectSpecification_Configurations
- deleteConfigurationsChilds

Copyright © All Rights Reserved

standard

Description

Standards: ISO, IRAM, ...

Properties

Name	Value
Engine	InnoDB
Auto Increment	7
Average Row Length	16384
Charset	utf8mb4
Collation	utf8mb4_general_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	24/6/2024 14:14:44

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	name	VARCHAR	60			False	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	False	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0

Triggers

Name	Type	Event	Security	Definer
deleteStandardChilds	BEFORE	DELETE	Definer	root

Copyright © All Rights Reserved

SQL Script

```sql
CREATE TABLE standard (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  name varchar(60) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
AVG_ROW_LENGTH = 16384,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_general_ci,
COMMENT = 'Standards: ISO, IRAM, ...';
```

Depends On

No items found

Used By 19

- selectMaterials
- selectStandards
- deleteStandard
- insertConditionalPeriod
- insertEndCap
- insertEnviroment
- insertRelatedMaterial
- insertStandard
- insertTestType
- selectStandardsJSON
- updateStandard
- conditional_period
- endcap
- enviroment
- material_has_standard
- deleteStandardChilds
- test_type
- selectStandarsComplete
- existStandard

Copyright © All Rights Reserved

test_type

Description

Properties

Name	Value
Engine	InnoDB
Auto Increment	0
Average Row Length	0
Charset	utf8mb4
Collation	utf8mb4_0900_ai_ci
Row Format	Dynamic
Min Rows	0
Max Rows	0
Checksum	False
Page Checksum	True
Pack Keys	False
Delay Key Write	False
Is Partitioned	False
Encryption	False
Persistent Statistics	DEFAULT
Auto Recalculate Statistics	DEFAULT
Sample Pages	0
Created	11/6/2024 11:06:02
Last Modified	24/6/2024 23:38:54

Columns

Key	Name	Data Type	Length	Precision	Scale	Unsigned	Zerofill	Binary	Not Null	Auto Increment	Default	Virtual	Invisible	Description
	id	INT		11		True	False	False	True	True		False	False	
	standard	INT		11		True	False	False	True	False		False	False	
	testType	VARCHAR	150			False	False	False	True	False		False	False	
	createdAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	
	updatedAt	DATETIME		0		False	False	False	True	False	CURRENT_TIMESTAMP	False	False	

Indexes

Key	Name	Columns	Unique	Type	Key Lengths
		id	True	None	0
	FK_test_type_standard_id	standard	False	None	0

Foreign Keys

Copyright © All Rights Reserved

Name	Columns	Delete Rule	Update Rule
FK_test_type_standard_id	id	N/S	N/S

SQL Script

```
CREATE TABLE test_type (
  id int UNSIGNED NOT NULL AUTO_INCREMENT,
  standard int UNSIGNED NOT NULL,
  testType varchar(150) NOT NULL,
  createdAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updatedAt datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
)
ENGINE = INNODB,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

ALTER TABLE test_type
ADD CONSTRAINT FK_test_type_standard_id FOREIGN KEY (standard)
REFERENCES standard (id) ON DELETE CASCADE ON UPDATE CASCADE;
```

Depends On 1

- standard

Used By 6

- selectTestType
- deleteTestType
- insertTestType
- existTest_Type
- selectTest_Type
- deleteStandardChilds

Copyright © All Rights Reserved

Procedures

Objects 38

Name	Description
deleteConditionalPeriod	
deleteEndCap	
deleteEnviroment	
deleteMaterial	
deleteOperator	
deleteRelatedMaterial	
deleteSpecification	
deleteSpecification_Configuration	
deleteStandard	
deleteTestType	
insertConditionalPeriod	Insertion of new Conditional Period
insertEndCap	
insertEnviroment	
insertMaterial	
insertOperator	
insertRelatedMaterial	
insertSpecification	
insertSpecification_Configuration	
insertStandard	
insertTestType	
selectConditionalPeriods	Selection of Conditional Periods by Standard
selectEndCaps	
selectEnviroment	
selectMaterials	
selectMaterialsJSON	
selectOperators	
selectOperatorsJSON	
selectSettingsSpecification	
selectSpecifications	
selectStandards	
selectStandardsJSON	
selectStandarsComplete	
selectTestType	

Copyright © All Rights Reserved

updateMaterial	
updateOperator	
updateSpecification	
updateSpecification_Configuration	
updateStandard	

Copyright © All Rights Reserved

deleteConditionalPeriod

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	24/6/2024 22:49:34
Last Modified	24/6/2024 22:49:34

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteConditionalPeriod (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existConditionalPeriod(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM conditional_period cp
    WHERE cp.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existConditionalPeriod
- conditional_period

Used By

No items found

Copyright © All Rights Reserved

deleteEndCap

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteEndCap (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existEndCap(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM endcap cp
    WHERE cp.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existEndCap
- endcap

Used By

No items found

Copyright © All Rights Reserved

deleteEnviroment

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteEnviroment (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existEnviroment(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM enviroment e
    WHERE e.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existEnviroment
- enviroment

Used By

No items found

Copyright © All Rights Reserved

deleteMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteMaterial (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existMaterial(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM material m
    WHERE m.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existMaterial
- material

Used By

No items found

Copyright © All Rights Reserved

deleteOperator

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteOperator (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existOperator(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM operator o
    WHERE o.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existOperator
- operator

Used By

No items found

Copyright © All Rights Reserved

deleteRelatedMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteRelatedMaterial (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existRelatedMaterial(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM material_has_standard mhs
    WHERE mhs.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existRelatedMaterial
- material_has_standard

Used By

No items found

Copyright © All Rights Reserved

deleteSpecification

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteSpecification (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existSpecification(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM specification s
    WHERE s.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existSpecification
- specification

Used By

No items found

Copyright © All Rights Reserved

deleteSpecification_Configuration

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteSpecification_Configuration (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existSpecification_Configuration(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM specification_configuration sc
    WHERE sc.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existSpecification_Configuration
- specification_configuration

Used By

No items found

Copyright © All Rights Reserved

deleteStandard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteStandard (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existStandard(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM standard s
    WHERE s.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existStandard
- standard

Used By

No items found

Copyright © All Rights Reserved

deleteTestType

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE deleteTestType (IN id int UNSIGNED)
DETERMINISTIC
BEGIN

  IF (existTest_Type(id) = TRUE) THEN
    DELETE LOW_PRIORITY QUICK
      FROM test_type tt
    WHERE tt.id = id
    LIMIT 1;

    SELECT
      'Successful Deleted!' AS `response`;
  ELSE
    SELECT
      'Unsuccessful Deleted!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- existTest_Type
- test_type

Used By

No items found

Copyright © All Rights Reserved

insertConditionalPeriod

Description

Insertion of new Conditional Period

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	24/6/2024 22:41:34
Last Modified	24/6/2024 22:41:34

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
minwall	INT		11	
maxwall	INT		11	
time	INT		11	
timeType	VARCHAR	3		
aproxTime	INT		11	
aproxType	VARCHAR	3		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertConditionalPeriod (IN idStandard int UNSIGNED, IN minwall int UNSIGNED, IN maxwall int UNSIGNED, IN
time int UNSIGNED, IN timeType varchar(3), IN aproxTime int UNSIGNED, IN aproxType varchar(3))
DETERMINISTIC
COMMENT 'Insertion of new Conditional Period'
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM conditional_period cp
  WHERE cp.standard = idStandard
  AND cp.minwall = minwall
  AND cp.maxwall = maxwall
  AND cp.time = time
  AND cp.timeType LIKE timeType
  AND cp.aproxTime = aproxTime
  AND cp.aproxType LIKE aproxType;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO conditional_period (standard, time, timeType, aproxTime, aproxType, minwall, maxwall)
```


Copyright © All Rights Reserved

```
      VALUES (idStandard, time, timeType, aproxTime, aproxType, minwall, maxwall);

    SELECT
      cp.id AS `key`,
      CONCAT(CONVERT(cp.time, char), ' ', cp.timeType, ' ± ', CONVERT(cp.aproxTime, char), '
', cp.aproxType) AS `condperiod`,
      cp.minwall AS `minwall`,
      cp.maxwall AS `maxwall`
    FROM conditional_period cp
    WHERE cp.standard = idStandard
    AND cp.minwall = minwall
    AND cp.maxwall = maxwall
    AND cp.time LIKE time;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- conditional_period
- standard

Used By

No items found

Copyright © All Rights Reserved

insertEndCap

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
endCap	VARBINARY	15		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertEndCap (IN idStandard int UNSIGNED, IN endCap varbinary(15))
DETERMINISTIC
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM endcap ec
  WHERE ec.name = endCap
  AND ec.name = endCap;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO endcap (standard, name)
      VALUES (idStandard, endCap);

    SELECT
      ec.id AS `key`,
      ec.name AS `endcap`
    FROM endcap ec
    WHERE ec.name = endCap
    AND ec.name = endCap;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;

END
```

Depends On 3

Copyright © All Rights Reserved

databaseManager@%

endcap

standard

Used By

No items found

Copyright © All Rights Reserved

insertEnviroment

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
insideFluid	VARCHAR	30		
outsideFluid	VARCHAR	30		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertEnviroment (IN idStandard int UNSIGNED, IN insideFluid varchar(30), IN outsideFluid varchar(30))
DETERMINISTIC
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM enviroment e
  WHERE e.standard = idStandard
  AND e.inside LIKE insideFluid
  AND e.outside LIKE outsideFluid;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO enviroment (standard, inside, outside)
      VALUES (idStandard, insideFluid, outsideFluid);

    SELECT
      e.id AS `key`,
      e.inside AS `insideFluid`,
      e.outside AS `outsideFluid`
    FROM enviroment e
    WHERE e.standard = idStandard
    AND e.inside LIKE insideFluid
    AND e.outside LIKE outsideFluid;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;
```


Copyright © All Rights Reserved

```
END
```

Depends On 3

- databaseManager@%
- enviroment
- standard

Used By

No items found

Copyright © All Rights Reserved

insertMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	25/6/2024 00:54:27
Last Modified	25/6/2024 00:54:27

Parameters

Name	Data Type	Length	Precision	Scale
nameIN	VARCHAR	20		
descriptionIN	VARCHAR	60		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertMaterial (IN nameIN varchar(20), IN descriptionIN varchar(60))
DETERMINISTIC
BEGIN

  DECLARE selected bigint;
  DECLARE result json;

  SELECT
    COUNT(*) INTO selected
  FROM material m
  WHERE m.name LIKE nameIN;

  IF (selected = 0) THEN
    INSERT INTO material (name, description)
      VALUES (nameIN, descriptionIN);

    SELECT
      m.id AS `key`,
      m.name AS `material`,
      m.description AS `description`,
      '[]' AS `specifications`
    FROM material m
    WHERE m.name = nameIN;

  ELSE
    SELECT
      'Already Exists!' AS `response`;
  END IF;

END
```


Copyright © All Rights Reserved

Depends On 2

databaseManager@%

material

Used By

No items found

Copyright © All Rights Reserved

insertOperator

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

Parameters

Name	Data Type	Length	Precision	Scale
dni	INT		11	
name	VARCHAR	20		
familyname	VARCHAR	30		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertOperator (IN dni int UNSIGNED, IN name varchar(20), IN familyname varchar(30))
DETERMINISTIC
BEGIN

  DECLARE selected bigint;

  SELECT
    COUNT(*) INTO selected
  FROM operator o
  WHERE o.dni = dni;

  IF (selected = 0) THEN
    INSERT INTO operator (dni, name, familyname)
      VALUES (dni, name, familyname);

    SELECT
      o.id AS `key`,
      o.dni AS `dni`,
      o.name AS `name`,
      o.familyname AS `familyName`
    FROM operator o
    WHERE o.dni = dni;

  ELSE
    SELECT
      'Already Exists!' AS `response`;
  END IF;

END
```


Copyright © All Rights Reserved

Depends On 2

- databaseManager@%
- operator

Used By

No items found

Copyright © All Rights Reserved

insertRelatedMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
idMaterial	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertRelatedMaterial (IN idStandard int UNSIGNED, IN idMaterial int UNSIGNED)
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM material_has_standard mhs
  WHERE mhs.material = idMaterial
  AND mhs.standard = idStandard;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO material_has_standard (material, standard)
      VALUES (idMaterial, idStandard);

    SELECT
      mhs.id AS `key`,
      m.id AS `idMaterial`,
      m.name AS `material`
    FROM material_has_standard mhs
      INNER JOIN material m
        ON mhs.material = m.id
    WHERE mhs.material = idMaterial
    AND mhs.standard = idStandard;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;

END
```


Copyright © All Rights Reserved

Depends On 4

- databaseManager@%
- material_has_standard
- material
- standard

Used By

No items found

Copyright © All Rights Reserved

insertSpecification

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	25/6/2024 00:54:16
Last Modified	25/6/2024 00:54:16

Parameters

Name	Data Type	Length	Precision	Scale
materialIN	INT		11	
nameIN	VARCHAR	20		
descriptionIN	VARCHAR	60		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertSpecification (IN materialIN int UNSIGNED, IN nameIN varchar(20), IN descriptionIN varchar(60))
DETERMINISTIC
BEGIN

  DECLARE selected bigint;

  SELECT
    COUNT(*) INTO selected
  FROM specification s
  WHERE s.name = nameIN
  AND s.material = materialIN;

  IF (selected = 0) THEN
    INSERT INTO specification (material, name, description)
      VALUES (materialIN, nameIN, descriptionIN);

    SELECT
      s.id AS `key`,
      s.name AS `specification`,
      s.description AS `description`,
      '[]' AS `configurations`
    FROM specification s
    WHERE s.name LIKE nameIN;

  ELSE
    SELECT
      'Already Exists!' AS `response`;
  END IF;

END
```


Copyright © All Rights Reserved

Depends On 3

- databaseManager@%
- specification
- material

Used By

No items found

Copyright © All Rights Reserved

insertSpecification_Configuration

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
specificationIN	INT		11	
timeIN	SMALLINT		6	
timeTypeIN	VARCHAR	1		
temperatureIN	TINYINT		4	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertSpecification_Configuration (IN specificationIN int UNSIGNED, IN timeIN smallint, IN timeTypeIN
varchar(1), IN temperatureIN tinyint)
DETERMINISTIC
BEGIN

  DECLARE selected bigint;

  SELECT
    COUNT(*) INTO selected
  FROM specification_configuration sc
  WHERE sc.specification = specificationIN
  AND sc.time = timeIN
  AND sc.temperature = temperatureIN;

  IF (selected = 0) THEN
    INSERT INTO specification_configuration (specification, time, timeType, temperature)
      VALUES (specificationIN, timeIN, timeTypeIN, temperatureIN);

    SELECT
      sc.id AS `key`,
      sc.time AS `time`,
      sc.timeType AS `timeType`,
      sc.temperature AS `temperature`
    FROM specification_configuration sc
    WHERE sc.specification = specificationIN
    AND sc.time = timeIN
    AND sc.temperature = temperatureIN;
```


Copyright © All Rights Reserved

```
  ELSE
    SELECT
      'Already Exists!' AS `response`;
  END IF;

END
```

Depends On 3

- databaseManager@%
- specification_configuration
- specification

Used By

No items found

Copyright © All Rights Reserved

insertStandard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
nameIn	VARCHAR	40		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertStandard (IN nameIn varchar(40))
DETERMINISTIC
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM standard s
  WHERE s.name = nameIn;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO standard (name)
      VALUES (nameIn);

    SELECT
      s.id AS `key`,
      s.name AS `standard`,
      '[]' AS `materials`,
      '[]' AS `enviroments`,
      '[]' AS `endCaps`,
      '[]' AS conditionalPeriods
    FROM standard s
    WHERE s.name LIKE nameIn;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;

END
```

Depends On 2

Copyright © All Rights Reserved

databaseManager@%

standard

Used By

No items found

Copyright © All Rights Reserved

insertTestType

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	13/6/2024 01:29:06
Last Modified	13/6/2024 01:29:06

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
testTypeIN	VARBINARY	255		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE insertTestType (IN idStandard int UNSIGNED, IN testTypeIN varbinary(255))
DETERMINISTIC
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM test_type tt
  WHERE tt.standard = idStandard
  AND tt.testType = testTypeIN;

  IF elements = 0 THEN
    INSERT HIGH_PRIORITY INTO test_type (standard, testType)
      VALUES (idStandard, testTypeIN);

    SELECT
      tt.id AS `key`,
      tt.testType AS `testtype`
    FROM test_type tt
    WHERE tt.standard = idStandard
    AND tt.testType = testTypeIN;

  ELSE
    SELECT
      'Already Exist!' AS `response`;
  END IF;

END
```

Depends On 3

Copyright © All Rights Reserved

databaseManager@%

test_type

standard

Used By

No items found

Copyright © All Rights Reserved

selectConditionalPeriods

Description

Selection of Conditional Periods by Standard

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	24/6/2024 20:39:58
Last Modified	24/6/2024 20:39:58

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectConditionalPeriods (IN idStandard int UNSIGNED)
DETERMINISTIC
COMMENT 'Selection of Conditional Periods by Standard'
BEGIN

  SELECT
    cp.id AS idCondPeriod,
    cp.minwall AS minWall,
    cp.maxwall AS maxWall,
    (CONCAT(CONVERT(cp.time, char), ' ', cp.timeType, ' ± ', CONVERT(cp.aproxTime, char), '
', cp.aproxType)) AS condPeriod
  FROM conditional_period cp
  WHERE cp.standard = idStandard
  ORDER BY cp.minwall;

END
```

Depends On 2

- dataCollector@%
- conditional_period

Used By

No items found

Copyright © All Rights Reserved

selectEndCaps

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	False
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectEndCaps (IN idStandard int UNSIGNED)
BEGIN

  SELECT
    e.id AS `id`,
    e.name AS `endCap`
  FROM endcap e
  WHERE e.standard = idStandard;

END
```

Depends On 2

dataCollector@%

endcap

Used By

No items found

Copyright © All Rights Reserved

selectEnviroment

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectEnviroment (IN idStandard int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    ehs.id,
    ehs.inside,
    ehs.outside
  FROM enviroment ehs
  WHERE ehs.standard = idStandard;

END
```

Depends On 2

dataCollector@%

enviroment

Used By

No items found

Copyright © All Rights Reserved

selectMaterials

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectMaterials (IN idStandard int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    mhs.material AS idMaterial,
    m.name AS materialName,
    m.description AS materialDescription,
    mhs.type AS materialType
  FROM material m
    INNER JOIN material_has_standard mhs
      ON m.id = mhs.material
  WHERE mhs.standard = standard;

END
```

Depends On 4

- dataCollector@%
- material
- material_has_standard
- standard

Used By

No items found

Copyright © All Rights Reserved

selectMaterialsJSON

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectMaterialsJSON ()
DETERMINISTIC
BEGIN

  SELECT
    JSON_ARRAYAGG(JSON_OBJECT('key', m.id,
    'material', m.name,
    'description', (IF(m.description IS NULL, 'Sin Descripción', m.description)),
    'specifications', (selectSpecifications(m.id)))) AS `materials`
  FROM material m;

END
```

Depends On 3

- databaseManager@%
- selectSpecifications
- material

Used By

No items found

Copyright © All Rights Reserved

selectOperators

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectOperators ()
DETERMINISTIC
BEGIN

  SELECT
    o.id,
    o.dni,
    o.name,
    o.familyname
  FROM operator o;

END
```

Depends On 2

- dataCollector@%
- operator

Used By

No items found

Copyright © All Rights Reserved

selectOperatorsJSON

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectOperatorsJSON ()
DETERMINISTIC
BEGIN

  SELECT
    JSON_ARRAYAGG(JSON_OBJECT('key', o.id,
    'dni', o.dni,
    'name', o.name,
    'familyName', o.familyname)) AS `operators`
  FROM operator o;

END
```

Depends On 2

databaseManager@%

operator

Used By

No items found

Copyright © All Rights Reserved

selectSettingsSpecification

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idSpecification	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectSettingsSpecification (IN idSpecification int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    sc.id AS idSetting,
    sc.time AS time,
    sc.temperature AS temperature
  FROM specification_configuration sc
  WHERE sc.specification = idSpecification;

END
```

Depends On 2

dataCollector@%

specification_configuration

Used By

No items found

Copyright © All Rights Reserved

selectSpecifications

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
idMaterial	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectSpecifications (IN idMaterial int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    s.id AS idSpecification,
    s.name AS nameSpecification,
    s.description AS descriptionSpecification
  FROM specification s
  WHERE s.material = idMaterial;

END
```

Depends On 2

- dataCollector@%
- specification

Used By 1

- selectMaterialsJSON

Copyright © All Rights Reserved

selectStandards

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectStandards ()
DETERMINISTIC
BEGIN

  SELECT
    s.id AS idStandard,
    s.name AS standardName
  FROM standard s;

END
```

Depends On 2

- dataCollector@%
- standard

Used By

No items found

Copyright © All Rights Reserved

selectStandardsJSON

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	12/6/2024 22:38:42
Last Modified	12/6/2024 22:38:42

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectStandardsJSON ()
DETERMINISTIC
BEGIN

  SELECT
    JSON_ARRAYAGG(JSON_OBJECT('key', s.id,
    'standard', s.name,
    'materials', (selectMaterials_has_Standard(s.id)),
    'enviroments', (selectEnviroments_has_Standard(s.id)),
    'endCaps', (selectEndCaps_has_Standard(s.id)),
    'conditionalPeriods', (selectConditionalPeriods_has_Standard(s.id)),
    'testTypes', (selectTest_Type(s.id)))) AS `standards`
  FROM standard s;

END
```

Depends On 7

- databaseManager@%
- selectMaterials_has_Standard
- selectEnviroments_has_Standard
- selectEndCaps_has_Standard
- selectConditionalPeriods_has_Standard
- selectTest_Type
- standard

Used By

No items found

Copyright © All Rights Reserved

selectStandarsComplete

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectStandarsComplete ()
DETERMINISTIC
BEGIN

  SELECT
    s.id,
    s.name,
    (SELECT
        JSON_ARRAYAGG(JSON_OBJECT('id', e.id,
        'endcap', e.name))
      FROM endcap e
      WHERE e.standard = s.id) AS `endCaps`,
    (SELECT
        JSON_ARRAYAGG(JSON_OBJECT('id', e.id,
        'insertFluid', e.inside,
        'outsideFluid', e.outside))
      FROM enviroment e
      WHERE e.standard = s.id) AS `enviroment`,
    (SELECT
        JSON_ARRAYAGG(JSON_OBJECT('id', cp.id,
        'minwall', cp.minwall,
        'maxwall', cp.maxwall,
        'time', cp.time))
      FROM conditional_period cp
      WHERE cp.standard = s.id) AS conditionalPeriod,
    (SELECT
        JSON_ARRAYAGG(JSON_OBJECT('id', mhs.id,
        'idMaterial', mhs.material,
        'material', m.name,
        'description', m.description))
      FROM material_has_standard mhs
        INNER JOIN material m
          ON mhs.material = m.id
      WHERE mhs.standard = s.id) AS relatedMaterial
  FROM standard s;

END
```

Depends On 6

Copyright © All Rights Reserved

endcap

enviroment

conditional_period

material_has_standard

material

standard

Used By

No items found

Copyright © All Rights Reserved

selectTestType

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE selectTestType (IN idStandard int UNSIGNED)
DETERMINISTIC
BEGIN

  SELECT
    tt.id,
    tt.testType
  FROM test_type tt
  WHERE tt.standard = idStandard;

END
```

Depends On 2

dataCollector@%

test_type

Used By

No items found

Copyright © All Rights Reserved

updateMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
nameIN	VARCHAR	20		
descriptionIN	VARCHAR	80		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateMaterial (IN id int UNSIGNED, IN nameIN varchar(20), IN descriptionIN varchar(80))
DETERMINISTIC
BEGIN

  DECLARE result text;

  IF (existMaterial(id) = TRUE) THEN
    UPDATE LOW_PRIORITY material m
    SET m.id = id,
        m.name = nameIN,
        m.description = descriptionIN
    WHERE m.id = id;

    SELECT
      'Successful Updated!' INTO result;
  ELSE
    SELECT
      'Unsucessful Updated!' INTO result;
  END IF;

  SELECT
    result AS `response`;

END
```

Depends On 3

- databaseManager@%
- existMaterial

Copyright © All Rights Reserved

material

Used By

No items found

Copyright © All Rights Reserved

updateOperator

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
dni	INT		11	
name	VARCHAR	50		
familyname	VARCHAR	50		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateOperator (IN id int UNSIGNED, IN dni int UNSIGNED, IN name varchar(50), IN familyname varchar(50))
DETERMINISTIC
BEGIN

  DECLARE result text;

  IF (existOperator(id) = TRUE) THEN
    UPDATE LOW_PRIORITY operator o
    SET o.dni = dni,
        o.name = name,
        o.familyname = familyname
    WHERE o.id = id;

    SELECT
      'Successful Updated!' INTO result;
  ELSE
    SELECT
      'Unsuccessful Updated!' INTO result;
  END IF;

  SELECT
    result AS response;

END
```

Depends On 3

Copyright © All Rights Reserved

databaseManager@%

existOperator

operator

Used By

No items found

Copyright © All Rights Reserved

updateSpecification

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
nameIN	VARCHAR	20		
descriptionIN	VARCHAR	80		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateSpecification (IN id int UNSIGNED, IN nameIN varchar(20), IN descriptionIN varchar(80))
DETERMINISTIC
BEGIN

  DECLARE result text;

  IF (existSpecification(id) = TRUE) THEN
    UPDATE LOW_PRIORITY specification s
    SET s.id = id,
        s.name = nameIN,
        s.description = descriptionIN
    WHERE s.id = id;

    SELECT
      'Successful Updated!' INTO result;
  ELSE
    SELECT
      'Unsuccessful Updated!' INTO result;
  END IF;

  SELECT
    result AS `response`;
END
```

Depends On 3

databaseManager@%

existSpecification

Copyright © All Rights Reserved

specification

Used By

No items found

Copyright © All Rights Reserved

updateSpecification_Configuration

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
timeIN	SMALLINT		6	
timeTypeIN	VARCHAR	1		
temperatureIN	TINYINT		4	

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateSpecification_Configuration (IN id int UNSIGNED, IN timeIN smallint, IN timeTypeIN varchar(1), IN
temperatureIN tinyint)
DETERMINISTIC
BEGIN

  DECLARE result text;

  IF (existSpecification_Configuration(id) = TRUE) THEN
    UPDATE LOW_PRIORITY specification_configuration sc
    SET sc.id = id,
        sc.time = timeIN,
        sc.timeType = timeTypeIN,
        sc.temperature = temperatureIN
    WHERE sc.id = id;

    SELECT
      'Successful Updated!' INTO result;
  ELSE
    SELECT
      'Unsuccessful Updated!' INTO result;
  END IF;

  SELECT
    result AS response;
END
```


Copyright © All Rights Reserved

Depends On 3

- databaseManager@%
- existSpecification_Configuration
- specification_configuration

Used By

No items found

Copyright © All Rights Reserved

updateStandard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
nameIN	VARCHAR	60		

SQL Script

```
CREATE
DEFINER = 'root'
PROCEDURE updateStandard (IN id int UNSIGNED, IN nameIN varchar(60))
DETERMINISTIC
BEGIN

  DECLARE result text;

  IF (existStandard(id) = TRUE) THEN
    UPDATE LOW_PRIORITY standard s
    SET s.id = id,
        s.name = nameIN
    WHERE s.id = id;

    SELECT
      'Successful Updated!' INTO result;
  ELSE
    SELECT
      'Unsucessful Updated!' INTO result;
  END IF;

  SELECT
    result AS `response`;

END
```

Depends On 3

- databaseManager@%
- existStandard
- standard

Copyright © All Rights Reserved

Used By

No items found

Copyright © All Rights Reserved

Functions

Objects 19

Name	Description
existConditionalPeriod	
existConfiguration	
existEndCap	
existEnviroment	
existMaterial	
existOperator	
existRelatedMaterial	
existSpecification	
existSpecification_Configuration	
existSpecimen	
existStandard	
existTest_Type	
selectConditionalPeriods_has_Standard	Selection of Conditional Periods of a Standard
selectEndCaps_has_Standard	
selectEnviroments_has_Standard	
selectMaterials_has_Standard	
selectSpecifications	
selectSpecification_Configurations	
selectTest_Type	

Copyright © All Rights Reserved

existConditionalPeriod

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existConditionalPeriod (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM conditional_period cp
  WHERE cp.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

conditional_period

Used By 1

deleteConditionalPeriod

Copyright © All Rights Reserved

existConfiguration

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existConfiguration (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM specification s
  WHERE s.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

specification

Used By

No items found

Copyright © All Rights Reserved

existEndCap

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existEndCap (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM endcap e
  WHERE e.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

endcap

Used By 1

deleteEndCap

Copyright © All Rights Reserved

existEnviroment

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existEnviroment (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM enviroment e
  WHERE e.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

enviroment

Used By 1

deleteEnviroment

Copyright © All Rights Reserved

existMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existMaterial (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM material m
  WHERE m.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

material

Used By 2

deleteMaterial

updateMaterial

Copyright © All Rights Reserved

existOperator

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:03
Last Modified	11/6/2024 11:06:03

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existOperator (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM operator o
  WHERE o.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

- operator

Used By 2

- deleteOperator
- updateOperator

Copyright © All Rights Reserved

existRelatedMaterial

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existRelatedMaterial (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM material_has_standard mhs
  WHERE mhs.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

material_has_standard

Used By 1

deleteRelatedMaterial

Copyright © All Rights Reserved

existSpecification

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existSpecification (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected int;
  SELECT
    COUNT(*) INTO selected
  FROM specification s
  WHERE s.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

- specification

Used By 2

- deleteSpecification
- updateSpecification

Copyright © All Rights Reserved

existSpecification_Configuration

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existSpecification_Configuration (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM specification_configuration sc
  WHERE sc.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

- specification_configuration

Used By 2

- deleteSpecification_Configuration
- updateSpecification_Configuration

Copyright © All Rights Reserved

Copyright © All Rights Reserved

existSpecimen

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:01
Last Modified	11/6/2024 11:06:01

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existSpecimen (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM specification s
  WHERE s.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

specification

Used By

No items found

Copyright © All Rights Reserved

existStandard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existStandard (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM standard s
  WHERE s.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

- standard

Used By 2

- deleteStandard
- updateStandard

Copyright © All Rights Reserved

existTest_Type

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
id	INT		11	
(Result)	TINYINT		1	

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION existTest_Type (id int UNSIGNED)
RETURNS tinyint(1)
DETERMINISTIC
BEGIN

  DECLARE selected bigint(255);
  SELECT
    COUNT(*) INTO selected
  FROM test_type tt
  WHERE tt.id = id;

  IF selected != 1 THEN
    RETURN FALSE;
  ELSE
    RETURN TRUE;
  END IF;

END
```

Depends On 1

test_type

Used By 1

deleteTestType

Copyright © All Rights Reserved

selectConditionalPeriods_has_Standard

Description

Selection of Conditional Periods of a Standard

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	24/6/2024 23:05:15
Last Modified	24/6/2024 23:05:15

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectConditionalPeriods_has_Standard (idStandard int UNSIGNED)
RETURNS json
DETERMINISTIC
COMMENT 'Selection of Conditional Periods of a Standard'
BEGIN

  DECLARE result json;
  DECLARE elements tinyint(1);

  SELECT
    COUNT(*) INTO elements
  FROM conditional_period cp
  WHERE cp.standard = idStandard;

  IF (elements > 0) THEN
    WITH periodsOrder
    AS
    (SELECT
        cp.id AS `key`,
        CONCAT(CONVERT(cp.time, char), ' ', cp.timeType, ' ± ', CONVERT(cp.aproxTime, char), ' ', cp.aproxType) AS
        `condPeriod`,
        cp.minwall,
        cp.maxwall
      FROM conditional_period cp
      WHERE cp.standard = idStandard)

    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', po.`key`,
      'condPeriod', po.condPeriod,
      'minwall', po.minwall,
      'maxwall', po.maxwall)) INTO result
```


Copyright © All Rights Reserved

```
    FROM periodsOrder po;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 1

- conditional_period

Used By 1

- selectStandardsJSON

Copyright © All Rights Reserved

selectEndCaps_has_Standard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectEndCaps_has_Standard (idStandard int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint(1);

  SELECT
    COUNT(*) INTO elements
  FROM endcap ec
  WHERE ec.standard = idStandard;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', ec.id,
      'endcap', ec.name)) INTO result
    FROM endcap ec
    WHERE ec.standard = idStandard;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 1

endcap

Copyright © All Rights Reserved

Used By 1

selectStandardsJSON

Copyright © All Rights Reserved

selectEnviroments_has_Standard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectEnviroments_has_Standard (idStandard int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint;

  SELECT
    COUNT(*) INTO elements
  FROM enviroment e
  WHERE e.standard = idStandard;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', e.id,
      'insideFluid', e.inside,
      'outsideFluid', e.outside)) INTO result
    FROM enviroment e
    WHERE e.standard = idStandard;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 1

Copyright © All Rights Reserved

enviroment

Used By 1

selectStandardsJSON

Copyright © All Rights Reserved

selectMaterials_has_Standard

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	24/6/2024 17:13:33
Last Modified	24/6/2024 17:13:33

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectMaterials_has_Standard (idStandard int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint;

  SELECT
    COUNT(*) INTO elements
  FROM material_has_standard mhs
  WHERE mhs.standard = idStandard;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', mhs.id,
      'idMaterial', mhs.material,
      'material', m.name,
      'descriptioin', m.description)) INTO result
    FROM material_has_standard mhs
      INNER JOIN material m
        ON m.id = mhs.material
    WHERE mhs.standard = idStandard;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```


Copyright © All Rights Reserved

Depends On 2

- material_has_standard
- material

Used By 1

- selectStandardsJSON

Copyright © All Rights Reserved

selectSpecifications

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idMaterial	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectSpecifications (idMaterial int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint;

  SELECT
    COUNT(*) INTO elements
  FROM specification s
  WHERE s.material = idMaterial;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', s.id,
      'specification', s.name,
      'description', (IF(s.description IS NULL, '', s.description)),
      'configurations', (selectSpecification_Configurations(s.id)))) INTO result
    FROM specification s
    WHERE s.material = idMaterial;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 2

Copyright © All Rights Reserved

specification

selectSpecification_Configurations

Used By

No items found

Copyright © All Rights Reserved

selectSpecification_Configurations

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	11/6/2024 11:06:02
Last Modified	11/6/2024 11:06:02

Parameters

Name	Data Type	Length	Precision	Scale
idSpecification	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectSpecification_Configurations (idSpecification int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint;

  SELECT
    COUNT(*) INTO elements
  FROM specification_configuration sc
  WHERE sc.specification = idSpecification;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', sc.id,
      'time', sc.time,
      'type', sc.timeType,
      'temperature', sc.temperature)) INTO result
    FROM specification_configuration sc
    WHERE sc.specification = idSpecification;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 1

Copyright © All Rights Reserved

specification_configuration

Used By 1

selectSpecifications

Copyright © All Rights Reserved

selectTest_Type

Description

Properties

Name	Value
SQL Access	CONTAINS SQL
Deterministic	True
Security	Definer
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Language	SQL
Created	25/6/2024 01:20:43
Last Modified	25/6/2024 01:20:43

Parameters

Name	Data Type	Length	Precision	Scale
idStandard	INT		11	
(Result)	JSON	0		

SQL Script

```
CREATE
DEFINER = 'root'
FUNCTION selectTest_Type (idStandard int UNSIGNED)
RETURNS json
DETERMINISTIC
BEGIN

  DECLARE result json;
  DECLARE elements tinyint;

  SELECT
    COUNT(*) INTO elements
  FROM test_type tt
  WHERE tt.standard = idStandard;

  IF (elements > 0) THEN
    SELECT
      JSON_ARRAYAGG(JSON_OBJECT('key', tt.id,
      'testtype', tt.testType)) INTO result
    FROM test_type tt
    WHERE tt.standard = idStandard;
  ELSE
    SELECT
      '[]' INTO result;
  END IF;

  RETURN result;
END
```

Depends On 1

test_type

Copyright © All Rights Reserved

Used By 1

selectStandardsJSON

Copyright © All Rights Reserved

Triggers

Objects 3

Name
deleteConfigurationsChilds
deleteMaterialChilds
deleteStandardChilds

Copyright © All Rights Reserved

deleteConfigurationsChilds

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	DELETE
Timing	BEFORE
Follows	
Precedes	
Created	11/6/2024 11:06:04

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER deleteConfigurationsChilds
BEFORE DELETE
ON specification
FOR EACH ROW
BEGIN

  DECLARE childsCount int UNSIGNED;

  SELECT
    COUNT(*) INTO childsCount
  FROM specification_configuration sc
  WHERE sc.specification = OLD.id;

  IF childsCount > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM specification_configuration sc
    WHERE sc.specification = OLD.id;

  END IF;

END
```

Depends On 2

- specification
- specification_configuration

Used By

No items found

Copyright © All Rights Reserved

deleteMaterialChilds

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	DELETE
Timing	BEFORE
Follows	
Precedes	
Created	11/6/2024 11:06:04

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER deleteMaterialChilds
BEFORE DELETE
ON material
FOR EACH ROW
BEGIN

  DECLARE childsCount int UNSIGNED;

  SELECT
    COUNT(*) INTO childsCount
  FROM specification s
  WHERE s.material = OLD.id;

  IF childsCount > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM specification s
    WHERE s.material = OLD.id;

  END IF;

  SELECT
    COUNT(*) INTO childsCount
  FROM material_has_standard mhs
  WHERE mhs.material = OLD.id;

  IF childsCount > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM material_has_standard mhs
    WHERE mhs.material = OLD.id;

  END IF;

END
```

Depends On 3

- material
- specification
- material_has_standard

Used By

No items found

Copyright © All Rights Reserved

Copyright © All Rights Reserved

deleteStandardChilds

Properties

Name	Value
Definer	root
SQL Mode	NO_AUTO_VALUE_ON_ZERO
Event	DELETE
Timing	BEFORE
Follows	
Precedes	
Created	11/6/2024 11:06:04

SQL Script

```
CREATE
DEFINER = 'root'
TRIGGER deleteStandardChilds
BEFORE DELETE
ON standard
FOR EACH ROW
BEGIN

  DECLARE elements int UNSIGNED;

  SELECT
    COUNT(*) INTO elements
  FROM endcap e
  WHERE e.standard = OLD.id;

  IF elements > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM endcap e
    WHERE e.standard = OLD.id;
  END IF;

  SELECT
    COUNT(*) INTO elements
  FROM conditional_period cp
  WHERE cp.standard = OLD.id;

  IF elements > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM conditional_period cp
    WHERE cp.standard = OLD.id;
  END IF;

  SELECT
    COUNT(*) INTO elements
  FROM enviroment e
  WHERE e.standard = OLD.id;

  IF elements > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM enviroment e
    WHERE e.standard = OLD.id;
  END IF;

  SELECT
    COUNT(*) INTO elements
  FROM test_type tt
  WHERE tt.standard = OLD.id;

  IF elements > 0 THEN
```


Copyright © All Rights Reserved

```
    DELETE LOW_PRIORITY QUICK
      FROM test_type tt
    WHERE tt.standard = OLD.id;
  END IF;

  SELECT
    COUNT(*) INTO elements
  FROM material_has_standard mhs
  WHERE mhs.standard = OLD.id;

  IF elements > 0 THEN
    DELETE LOW_PRIORITY QUICK
      FROM material_has_standard mhs
    WHERE mhs.standard = OLD.id;
  END IF;

END
```

Depends On 6

- standard
- endcap
- conditional_period
- enviroment
- test_type
- material_has_standard

Used By

No items found

Copyright © All Rights Reserved

Users

Objects 3

- dataCollector@%
- databaseManager@%
- reportGenerator@%

Copyright © All Rights Reserved

dataCollector@%

Object Level Permissions 16

Type	Action	Object Name	Object Type
Grant	EXECUTE	countspecimens	Procedure
Grant	EXECUTE	insertdata	Procedure
Grant	EXECUTE	insertsample	Procedure
Grant	EXECUTE	insertspecimen	Procedure
Grant	EXECUTE	selectconditionalperiods	Procedure
Grant	EXECUTE	selectendcaps	Procedure
Grant	EXECUTE	selectenviroment	Procedure
Grant	EXECUTE	selectmaterials	Procedure
Grant	EXECUTE	selectoperators	Procedure
Grant	EXECUTE	selectsample	Procedure
Grant	EXECUTE	selectsampleid	Procedure
Grant	EXECUTE	selectsettingsspecification	Procedure
Grant	EXECUTE	selectspecifications	Procedure
Grant	EXECUTE	selectspecimen	Procedure
Grant	EXECUTE	selectstandards	Procedure
Grant	EXECUTE	selecttesttype	Procedure

SQL Script

```
CREATE USER 'dataCollector'@'%' IDENTIFIED WITH caching_sha2_password PASSWORD EXPIRE NEVER;
GRANT EXECUTE ON PROCEDURE data_db.countspecimens TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.insertdata TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.insertsample TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.insertspecimen TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectconditionalperiods TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectendcaps TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectenviroment TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectmaterials TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectoperators TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selectsample TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selectsampleid TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectsettingsspecification TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectspecifications TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selectspecimen TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectstandards TO 'dataCollector'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selecttesttype TO 'dataCollector'@'%';
```


Copyright © All Rights Reserved

databaseManager@%

Object Level Permissions 29

Type	Action	Object Name	Object Type
Grant	EXECUTE	deleteconditionalperiod	Procedure
Grant	EXECUTE	deleteendcap	Procedure
Grant	EXECUTE	deleteenviroment	Procedure
Grant	EXECUTE	deletematerial	Procedure
Grant	EXECUTE	deleteoperator	Procedure
Grant	EXECUTE	deleterelatedmaterial	Procedure
Grant	EXECUTE	deletespecification_configuration	Procedure
Grant	EXECUTE	deletespecification	Procedure
Grant	EXECUTE	deletestandard	Procedure
Grant	EXECUTE	deletetesttype	Procedure
Grant	EXECUTE	insertconditionalperiod	Procedure
Grant	EXECUTE	insertendcap	Procedure
Grant	EXECUTE	insertenviroment	Procedure
Grant	EXECUTE	insertmaterial	Procedure
Grant	EXECUTE	insertoperator	Procedure
Grant	EXECUTE	insertrelatedmaterial	Procedure
Grant	EXECUTE	insertspecification_configuration	Procedure
Grant	EXECUTE	insertspecification	Procedure
Grant	EXECUTE	insertstandard	Procedure
Grant	EXECUTE	inserttesttype	Procedure
Grant	EXECUTE	selectmaterialsjson	Procedure
Grant	EXECUTE	selectoperatorsjson	Procedure
Grant	EXECUTE	selectstandardsjson	Procedure
Grant	EXECUTE	updatematerial	Procedure
Grant	EXECUTE	updateoperator	Procedure
Grant	EXECUTE	updatespecification_configuration	Procedure
Grant	EXECUTE	updatespecification	Procedure
Grant	EXECUTE	updatestandard	Procedure
Grant	SELECT	material	Table

SQL Script

```
CREATE USER 'databaseManager'@'%' IDENTIFIED WITH caching_sha2_password PASSWORD EXPIRE NEVER;
GRANT EXECUTE ON PROCEDURE static_db.deleteconditionalperiod TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deleteendcap TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deleteenviroment TO 'databaseManager'@'%';
```


Copyright © All Rights Reserved

```
GRANT EXECUTE ON PROCEDURE static_db.deletematerial TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deleteoperator TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deleterelatedmaterial TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deletespecification TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deletespecification_configuration TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deletestandard TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.deletetesttype TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertconditionalperiod TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertendcap TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertenviroment TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertmaterial TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertoperator TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertrelatedmaterial TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertspecification TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertspecification_configuration TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.insertstandard TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.inserttesttype TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectmaterialsjson TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectoperatorsjson TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.selectstandardsjson TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.updatematerial TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.updateoperator TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.updatespecification TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.updatespecification_configuration TO 'databaseManager'@'%';
GRANT EXECUTE ON PROCEDURE static_db.updatestandard TO 'databaseManager'@'%';
GRANT SELECT ON TABLE static_db.material TO 'databaseManager'@'%';
```


Copyright © All Rights Reserved

reportGenerator@%

Object Level Permissions 7

Type	Action	Object Name	Object Type
Grant	EXECUTE	static_db	Database
Grant	EXECUTE	deletetest	Procedure
Grant	EXECUTE	selectcomparetests	Procedure
Grant	EXECUTE	selecttestdata	Procedure
Grant	EXECUTE	selecttest	Procedure
Grant	EXECUTE	selecttests	Procedure
Grant	EXECUTE	updatespecimen	Procedure

SQL Script

```
CREATE USER 'reportGenerator'@'%' IDENTIFIED WITH caching_sha2_password PASSWORD EXPIRE NEVER;
GRANT EXECUTE ON *.* TO 'reportGenerator'@'%';
GRANT EXECUTE ON static_db.* TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.deletetest TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selectcomparetests TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selecttest TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selecttestdata TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.selecttests TO 'reportGenerator'@'%';
GRANT EXECUTE ON PROCEDURE data_db.updatespecimen TO 'reportGenerator'@'%';
```


Copyright © All Rights Reserved