

2021

YouTestMe

Querying test results from the YouTestMe
database

| | |
|------------------------|---|
| File name | Querying Test Results From The Youtestme Database |
| Author | YouTestMe |
| Confidentiality | Internal |
| Last save date | Tuesday, June-08-2021 at 2:23:00 PM |

Table of Contents

| | | |
|---|--|---|
| 1 | Introduction..... | 2 |
| 2 | Querying test attempt results | 2 |
| 3 | Querying test attempt results by the finishing (completion) time | 4 |
| 4 | Querying test attempt results by the last modification time | 5 |
| 5 | Querying test attempt responses for a test without sections..... | 6 |
| 6 | Querying test attempt responses for a test without sections..... | 7 |

1 Introduction

This document describes how to query the test attempt results from the YouTestMe database.

2 Querying test attempt results

The SQL view **V_QUIZ_INSTANCE_RESULT** contains all data of the candidate test attempts and their results. The following SQL query selects only the most important columns for the results of the test attempts from the **V_QUIZ_INSTANCE_RESULT** view :

```

SELECT user_id,
       username,
       first_name,
       last_name,
       email,
       quiz_definition_id,
       quiz_name,
       quiz_instance_id,
       quiz_started,
       quiz_finished,
       quiz_instance_status,
       max_points,
       points_assigned,
       percent_scored,
       passed_indicator,
       instance_update_datetime
FROM v_quiz_instance_result
WHERE lead_quiz_instance_id IS NULL;

```

The **WHERE** clause excludes the test section attempts.

The table below lists all the columns of the query above:

| Column | Type | Can be null | Description |
|--------------------------|--------------|-------------|---|
| user_id | numeric(10) | no | The ID of the candidate |
| username | varchar(900) | no | The username of the candidate |
| first_name | varchar(150) | no | The first name of the candidate |
| last_name | varchar(150) | no | The last name of the candidate |
| email | varchar(385) | no | The email of the candidate |
| quiz_definition_id | numeric(10) | no | The ID of the test |
| quiz_name | varchar(300) | no | The name of the test |
| quiz_instance_id | numeric(10) | no | The ID of the test attempt, unique for all records from the view |
| quiz_started | timestamp | yes | The time when the test attempt is started |
| quiz_finished | timestamp | yes | The time when the test attempt is finished |
| quiz_instance_status | varchar(10) | no | Status of the attempt Possible values: NAT – Not attempted ATN – Attempted, the report is not ready yet ATR – Attempted, the report is ready QSS – Suspended attempt AFC – Attempt is available for continuation by the candidate PRV – Preview attempt, taken by instructor |
| max_points | numeric(10) | yes | Total points on the test |
| points_assigned | numeric(10) | yes | The number of points candidate achieved on the test |
| percent_scored | numeric | yes | Candidate's percentage score on the test (points_assigned/max_points*100) |
| passed_indicator | char(1) | yes | 'Y' indicates that the candidate passed the test, 'N' indicates that the candidate failed |
| instance_update_datetime | timestamp | yes | The time of the last update of the test attempt |

The table below lists the combination of values of different attempt parameters and their meanings and expected actions:

| quiz_instance_status | quiz_started | quiz_finished | Meaning |
|----------------------|-------------------|---------------|---|
| NAT | NULL (always) | NULL (always) | The candidate has not attempted the test yet. |
| QSS | ANY | ANY | The test attempt is suspended. The attempt is considered archived and immutable. Typically there is a new attempt for the candidate on the same test. |
| PRV | ANY | ANY | Represents a temporary attempt record only when the instructor reviews the test to see if everything is set up properly. |
| AFC | NOT NULL (always) | NULL (always) | When the candidate leaves or pauses, attempt to continue it later or when the instructor allows him to continue. |
| ATR OR ATN | NOT NULL | NULL | The candidate has started the test attempt, and it is currently in progress. |
| ATR | NOT NULL | NOT NULL | The candidate has finished the test and can see the report. The instructor can still change the grade. |
| ATN | NOT NULL | NOT NULL | The candidate has finished the test but can't see the report. The instructor has to do the manual grading or approve the final grade for preview (this the setting in a test that allows the instructor to withhold the preview of the results until they are ready). |

3 Querying test attempt results by the finishing (completion) time

The following SQL query selects all test attempts finished (completed) after 2021/04/15:

```
SELECT user_id,
       username,
       first_name,
       last_name,
       email,
       quiz_definition_id,
       quiz_name,
       quiz_instance_id,
       quiz_started,
       quiz_finished,
       quiz_instance_status,
       max_points,
       points_assigned,
       percent_scored,
       passed_indicator,
       instance_update_datetime
FROM v_quiz_instance_result
WHERE lead_quiz_instance_id IS NULL
      AND quiz_finished IS NOT NULL
      AND quiz_finished >= '2021-04-15';
```

Still, the results of the test attempts returned by this SQL query can be changed. For example, if the instructor still has to grade some questions that the system can't grade automatically. So, the query above will not list those changes, which means that it is unsuitable for some periodical synchronization of the corresponding records in the external database. The query is suitable only if the grade data (points_assigned, percent_scored, and passed_indicator) is not saved in the external database.

4 Querying test attempt results by the last modification time

The following SQL query selects all finished (completed) test attempts and modified after 2021/04/15 for the last time:

```
SELECT user_id,
       username,
       first_name,
       last_name,
       email,
       quiz_definition_id,
       quiz_name,
       quiz_instance_id,
       quiz_started,
       quiz_finished,
       quiz_instance_status,
       max_points,
       points_assigned,
       percent_scored,
       passed_indicator,
       instance_update_datetime
FROM v_quiz_instance_result
WHERE lead_quiz_instance_id IS NULL
      AND quiz_finished IS NOT NULL
      AND instance_update_datetime >= '2021-04-15';
```

The query above will list attempts with the changes in grade data(points_assigned, percent_scored, and passed_indicator). So it is more suitable if the grade data is captured in the external database. Test attempt records in the external database should be updateable based on the quiz_instance_id value (ID of the test attempt in the YouTestMe database).

For example, let's say that synchronization with the external database is executed on 2021/04/15, and the next execution time is on 2021/05/15:

Synchronization on 2021/04/15:

The record with quiz_instance_id = 100609 is retrieved for the first time and should be inserted into the external database:

```
(1,"adis.dijab","Adis","Dijab","adis.dijab@youtestme.com",100064,"Security certification",100609,"2021-02-24 10:45:44.923","2021-04-29 15:35:17.85","ATN",60,34,56.6666666666666666700,"N","2021-04-29 15:36:32.924268")
```

Synchronization on 2021/05/15:

The record with quiz_instance_id = 100609 is retrieved again, and changes should be updated into the external database:

```
(1,"adis.dijab","Adis","Dijab","adis.dijab@youtestme.com",100064,"Security certification",100609,"2021-02-24 10:45:44.923","2021-04-29 15:35:17.85","ATR",60,45,75,"Y","2021-05-01 15:36:32.924268")
```

5 Querying test attempt responses for a test without sections

The following SQL query selects the candidate's responses to the questions on a test without sections:

```
SELECT overall_quiz_id ,
       overall_quiz_name ,
       quiz_instance_id ,
       display_ordinal ,
       question_id ,
       question_type ,
       question_text ,
       answer_text ,
       response ,
       points_assigned ,
       points ,
       correct_answer
FROM v_rep_user_no_sections_question_result
WHERE quiz_instance_id = 3780
ORDER BY display_ordinal;
```

The query filters all responses by attempt ID and orders them in the way they appeared on the attempt.

The candidate response is correct if the points_assigned are equal to points.

The table below lists all the columns of the query above:

| Column | Type | Can be null | Description |
|-------------------|--------------|-------------|-----------------------|
| overall_quiz_id | numeric(10) | no | The ID of the test |
| overall_quiz_name | varchar(300) | no | The name of the test |
| quiz_instance_id | numeric(10) | no | The ID of the attempt |

| | | | |
|-----------------|-------------|-----|---|
| display_ordinal | numeric(10) | no | The index of the question on the attempt |
| question_id | numeric(10) | no | The ID of the question |
| question_type | varchar(10) | no | The code of the question type: SNC – single choice, MLC – multiple choice, TFC – true/false, ESY – essay, ORD – ordering, FBL – fill in the blanks, MCH – matching, OPQ – open text, SUR – survey, MTR – matrix, HTS – hotspot |
| question_text | text | no | The text of the question |
| answer_text | text | yes | Free text response on essay or open text questions |
| response | text | yes | Comma-separated indexes of the answers selected for a single choice, multiple choice, or true/false questions. |
| points_assigned | numeric | yes | The points assigned for the response |
| points | integer | no | Total points question carries |
| correct_answer | text | yes | Comma-separated indexes of the correct answers |

6 Querying test attempt responses for a test with sections

The following SQL query selects the candidate's responses to the questions on a test with sections:

```

SELECT overall_quiz_id ,
       overall_quiz_name ,
       section_quiz_id ,
       section_name ,
       lead_inst_id ,
       display_ordinal ,
       question_id ,
       question_type ,
       question_text ,
       answer_text ,
       response ,
       points_assigned ,
       points ,
       correct_answer
FROM v_rep_user_section_question_result
WHERE lead_inst_id = 4064
ORDER BY section_quiz_id ,
         display_ordinal ;

```

The query filters all responses by attempt ID and orders them in the way they appeared on the attempt sections. The candidate response is correct if the points_assigned are equal to points.

The table below lists all the columns of the query above:

| Column | Type | Can be null | Description |
|-------------------|--------------|-------------|---|
| overall_quiz_id | numeric(10) | no | The ID of the test |
| overall_quiz_name | varchar(300) | no | The name of the test |
| section_quiz_id | numeric(10) | no | The ID of the test section |
| section_name | varchar(300) | no | The name of the test section |
| lead_inst_id | numeric(10) | no | The ID of the attempt |
| display_ordinal | numeric(10) | no | The index of the question on the attempt |
| question_id | numeric(10) | no | The ID of the question |
| question_type | varchar(10) | no | The code of the question type: SNC – single choice, MLC – multiple choice, TFC – true/false, ESY – essay, ORD – ordering, FBL – fill in the blanks, MCH – matching, OPQ – open text, SUR – survey, MTR – matrix, HTS – hotspot |
| question_text | text | no | The text of the question |
| answer_text | text | yes | Free text response on essay or open text questions |
| response | text | yes | Comma-separated indexes of the answers selected for a single choice, multiple choice, or true/false questions. |
| points_assigned | numeric | yes | The points assigned for the response |
| points | integer | no | Total points question carries |
| correnct_answer | text | yes | Comma-separated indexes of the correct answers |