**Assignment-3**

|  |  |
| --- | --- |
| **Student Name** | Wildan Luqmanul Hakim |
| **Unit Number and Title** | ACWD Module 4 – Database Design & Implementation |
| **Academic Year** | 2022 |
| **Unit Assessor** | Arvinder Kaur |
| **Project Title** | Implement a Database Design for Community Portal |
| **Issue Date** | 29 July 2022 |
| **Submission Date** | 2 August 2022 |
| **Internal Verifier Name** |  |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.    Student signature: Date: 2 August 2022 |

**Submission Format**

1. Screen Capture of Sample Data for Report
2. Queries for generating reports from database
3. Detailed Description of Test Methods
4. Documentation of database

**The scope of this assignment**

1. Create test data for all tables suitable for generating reports
2. Design report and create queries which will be used for generation of meaningful management reports. Present the report in HTML format (1 report)
3. Discuss briefly test methods you will employ to test and validate the database and brief reason why choose each test
4. Document database
5. Create a batch script to backup database & schedule it to run every 6 hours using windows task scheduler. Provide the script, along with the screen capture of Windows Task Scheduler
6. Provide a Restoration script in case of failure
7. **Create test data for all tables suitable for generating reports**

|  |  |
| --- | --- |
| **Table** | user |
| **SQL Query** | INSERT INTO user  (`user\_id`, `first\_name`, `last\_name`, `company`, `city`, `country`, `email`, `password`) VALUES  ('1', 'Barry', 'Portillo', 'Google', 'Jakarta ', 'Indonesia', 'barry@abc.com', 'barry123'),  ('2', 'Jannat', 'Hamilton', 'Google', 'Tokyo', 'Japan', 'jannat@abc.com', 'jannat123'),  ('3', 'Aniyah', 'Hayden', 'Samsung', 'Bandung', 'Indonesia', 'aniyah@abc.com', 'aniyah123'),  ('4', 'Elliot', 'Hogan', 'Amazon', 'New York', 'USA', 'elliot@abc.com', 'elliot123'),  ('5', 'Howard', 'Richards', 'Microsoft', 'Berlin', 'German', 'howard@abc.com', 'howard123'),  ('6', 'Wildan', 'Luqmanul Hakim', 'Google', 'Bandung', 'Indonesia', 'wildanluqmanul@gmail.com', 'password123'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | user\_profile |
| **SQL Query** | INSERT INTO user\_profile  (`user\_id`, `status\_id`, `username`, `job\_status`, `experience`, `company`, `bio`)  VALUES  ('1', '1', 'barry\_p', 'Working', 'Sofware Engineer ', 'Google', 'Lorem ipsum dolor sit amet'),  ('2', '2', 'jnnath', 'Working', 'UI/UX Design', 'Google', 'Lorem ipsum dolor sit amet'),  ('3', '3', 'aniyyyaah', 'Working', 'Back End Developer', 'Samsung', 'Lorem ipsum dolor sit amet'),  ('4', '4', 'elliot67', 'Working', 'Business Analysis', 'Amazon', 'Lorem ipsum dolor sit amet'),  ('5', '5', 'howrdddd', 'Working', 'Problem Solving', 'Microsoft', 'Lorem ipsum dolor sit amet'),  ('6', '6', 'wildanlh', 'Working', 'Front End Developer', 'Google', 'Lorem ipsum dolor sit amet'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | admin |
| **SQL Query** | INSERT INTO admin  (`admin\_id`, `first\_name`, `last\_name`, `email`, `password`)  VALUES  ('1', 'Parker ', 'Paterson', 'parker@adminabc.com', 'parker123'),  ('2', 'Raphael ', 'Crosby', 'raphael@adminabc.com', 'raphael123'),  ('3', 'Luqman', 'Hakim', 'luqman@adminabc.com', 'luqman123'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | company |
| **SQL Query** | INSERT INTO company  (`company\_id`, `company\_name`, `company\_address`)  VALUES  ('1', 'Amazon', 'Seattle, Washington and Arlington, Virginia, U.S.'),  ('2', 'Google', '1600 Amphitheatre Parkway, Mountain View, California, U.S.'),  ('3', 'Nvidia', 'Santa Clara, California, U.S.'),  ('4', 'Meta', 'Menlo Park, California, U.S.'),  ('5', 'Oracle', 'Austin, Texas, United States'),  ('6', 'Intel', 'Santa Clara, California, U.S.'),  ('7', 'Microsoft', 'One Microsoft Way Redmond, Washington, U.S.'),  ('8', 'Samsung', '40th floor Samsung Electronics Building, 11, Seocho-daero 74-gil, Seocho District, Seoul, South Korea'),  ('9', 'Tencent', 'Tencent Binhai Mansion, Nanshan District, Shenzhen, Guangdong, China'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | job |
| **SQL Query** | INSERT INTO job  (`job\_id`, `company\_name`, `job\_desc`, `job\_type`)  VALUES  ('1', ' Google', 'We are looking for Back End Developer', 'Full-Time'),  ('2', 'Amazon', 'We are looking for Fullstack Developer', 'Part-Time'),  ('3', 'Microsoft', 'We are looking for UI Designer', 'Full-Time'),  ('4', 'Samsung', 'We are looking for Front End Developer', 'Full-Time'),  ('5', 'Nvidia', 'We are looking for Software Engineer', 'Part-Time'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | thread |
| **SQL Query** | INSERT INTO `thread`  (`thread\_id`, `date`, `time`, `thread\_desc`, `comment`)  VALUES  ('1', '2021-02-09', '10:34 PM', 'Very Happy', 'How was your day?'),  ('2', '2021-11-11', '09:52 AM', 'Its time to coffe', 'Morning with coffe'),  ('3', '2022-01-23', '07:32 AM', 'Im working on Google ', 'Google is the best company'),  ('4', '2022-03-01', '03:00 PM', 'The best experience ', 'Whats happen today?'),  ('5', '2022-04-19', '08:03 AM', 'I like this Nvidia Company ', 'What about update on RTX Graphic Card?'); |
| **Screenshot** |  |

|  |  |
| --- | --- |
| **Table** | message |
| **SQL Query** | INSERT INTO `message`  (`message\_id`, `date`, `time`, `message`, `to`, `from`)  VALUES  ('1', '2021-08-01', '10:34 AM', 'Good Morning sir?', 'Barry Portillo', 'Elliot Hogan'),  ('2', '2022-08-01', '10:35 AM', 'Morning sir', 'Elliot Hogan', 'Barry Portillo'),  ('3', '2022-08-02', '09:00 AM', 'How are you?', 'Jannat Hamilton', 'Aniyah Hayden'),  ('4', '2022-08-02', '11:35 AM', 'Im quite fine, How about you?', 'Aniyah Hayden', 'Jannat Hamilton'),  ('5', '2022-08-02', '12:35 PM', 'Thanks', 'Howard Richards', 'Barry Portillo'); |
| **Screenshot** |  |

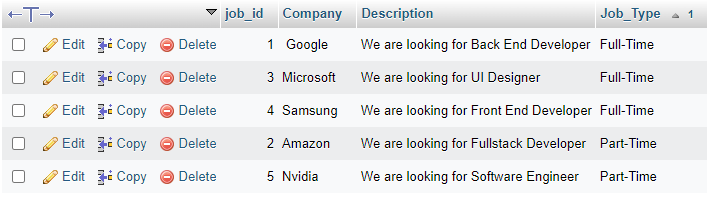
1. **Design report and create queries which will be used for generation of meaningful management reports.**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Note | Query | Evidence |
| 1 | All users contact info.  To fetch users, contact info information | SELECT user\_id, CONCAT(first\_name, " ", last\_name) AS Name, CONCAT(city,", ",country) AS Location, email AS Email  FROM user  ORDER BY Name; | Report users contact log |
| 2 | All jobs type info, To fetch job type information | SELECT job\_id, company\_name AS Company, job\_desc AS Description, job\_type AS Job\_Type  FROM job  ORDER BY Job\_Type; | Report jobs type log |

**Report users contact log**



**Report job type log**

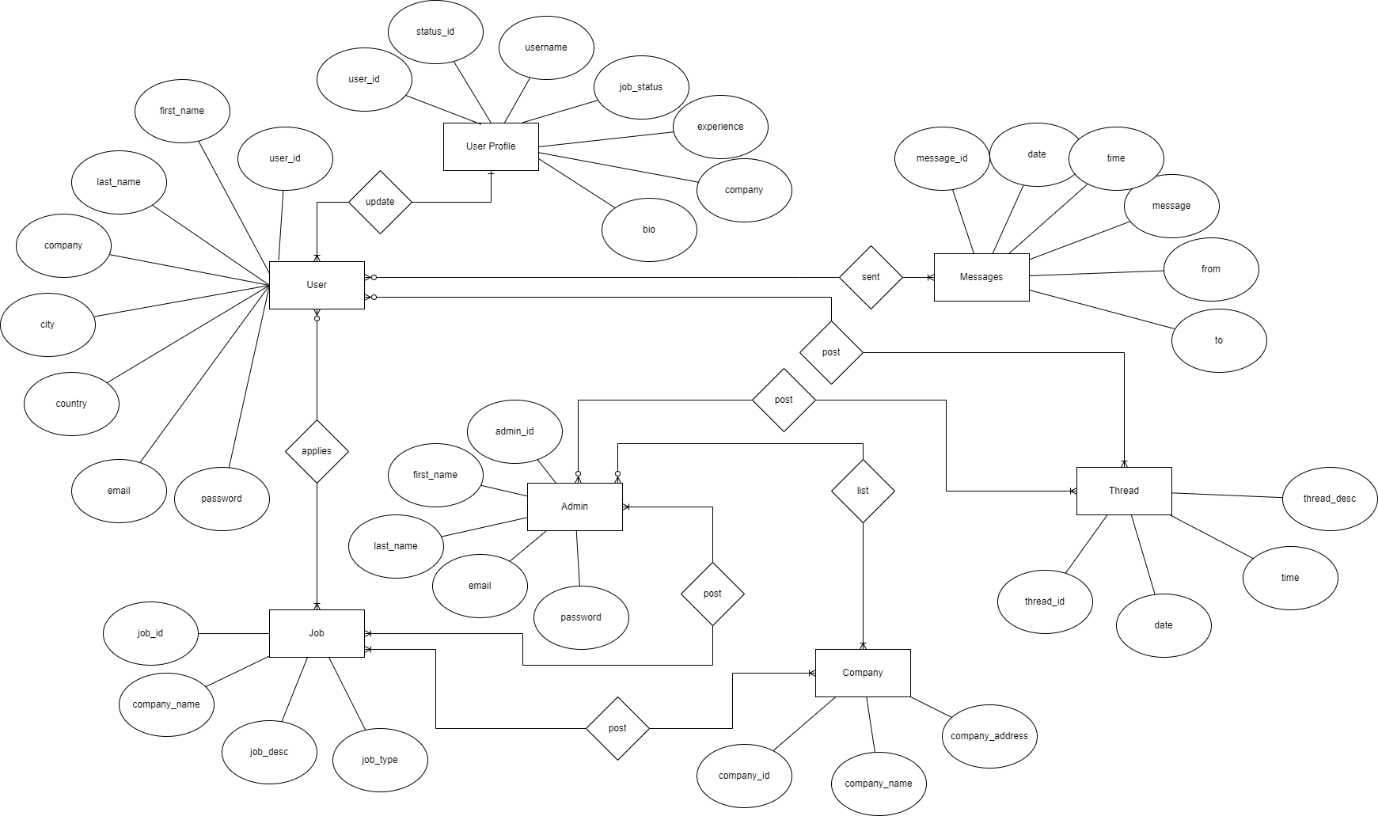


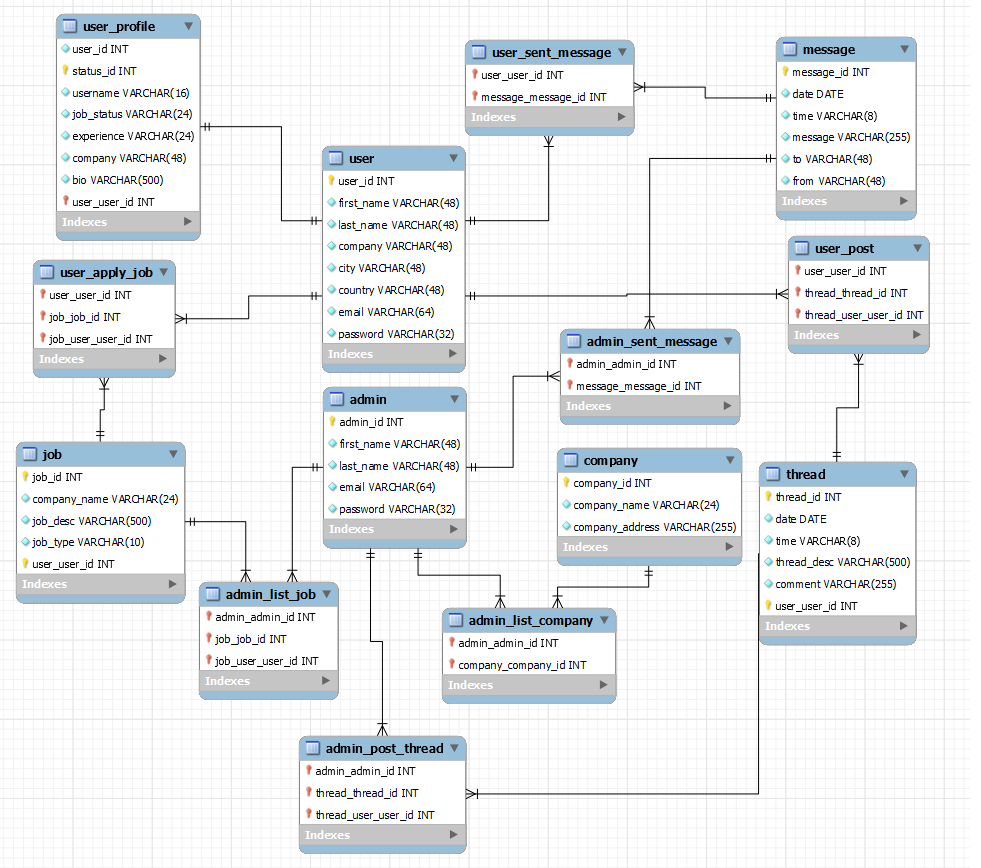
1. **Discuss briefly test methods you will employ to test and validate the database and brief reason why choose each test**
2. **Structural Database Testing**
   1. **Schema Testing.**

Schema or database testing is vital in ensuring the validity of data received and stored into database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. / Test Cases | Expected Result | Actual Result | Test Result | Evidence |
| 1. Schema Testing: Ensure that Relationship Schema and EER Diagram is the same in terms of (table name, attributes, primary key and foreign key) | Both Relationship schema and EER diagram should have similar field of: table name, attributes, primary key and foreign key | Both Relationship schema and EER diagram is the same in terms of (table name, attributes, primary key and foreign key)  \*See diagram in next page | Pass | Screen capture of logical design and physical design (EERD) |

**Evidence:**

Relationship Schema  


EER Diagram  


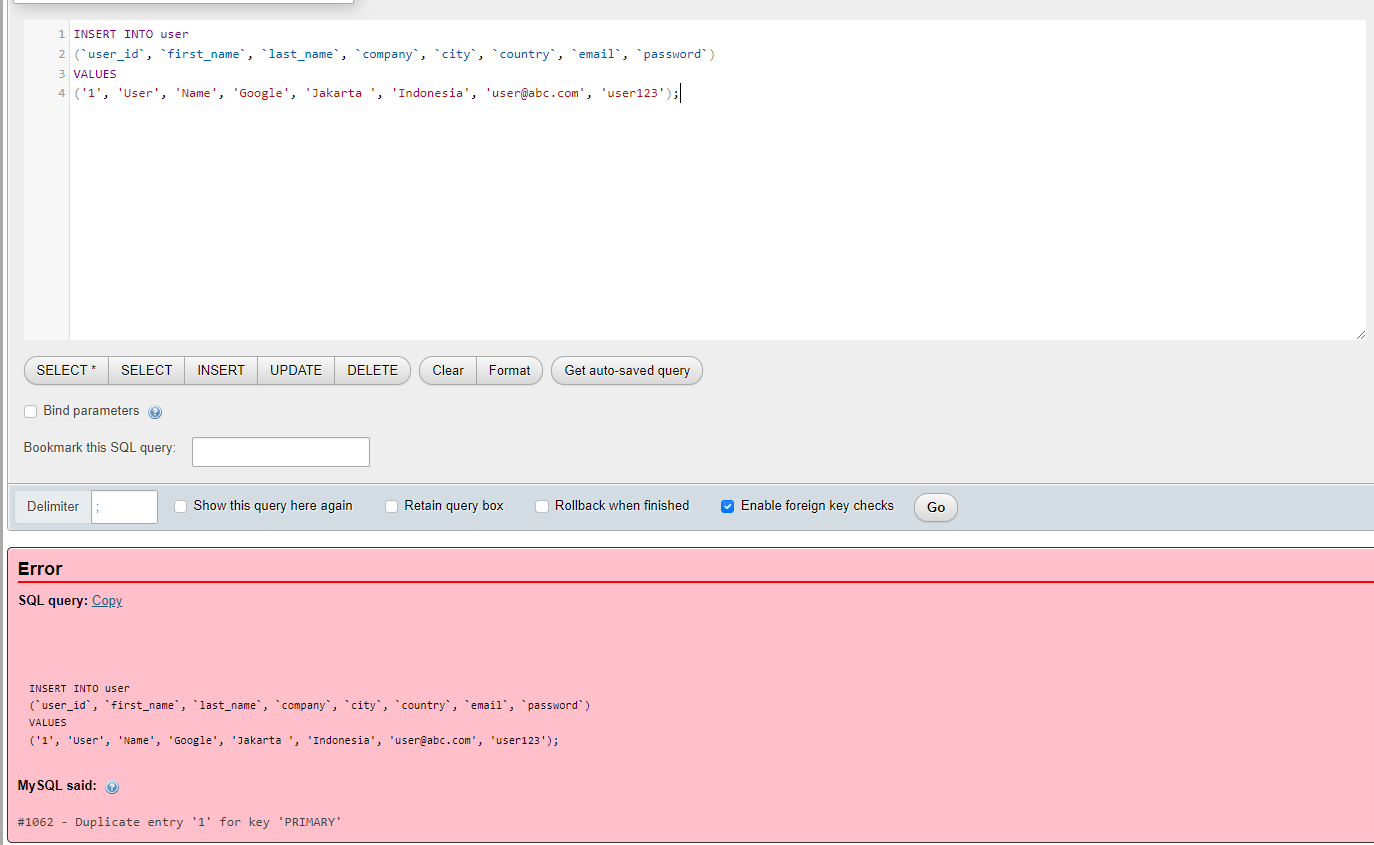
* 1. **Table/column testing.**

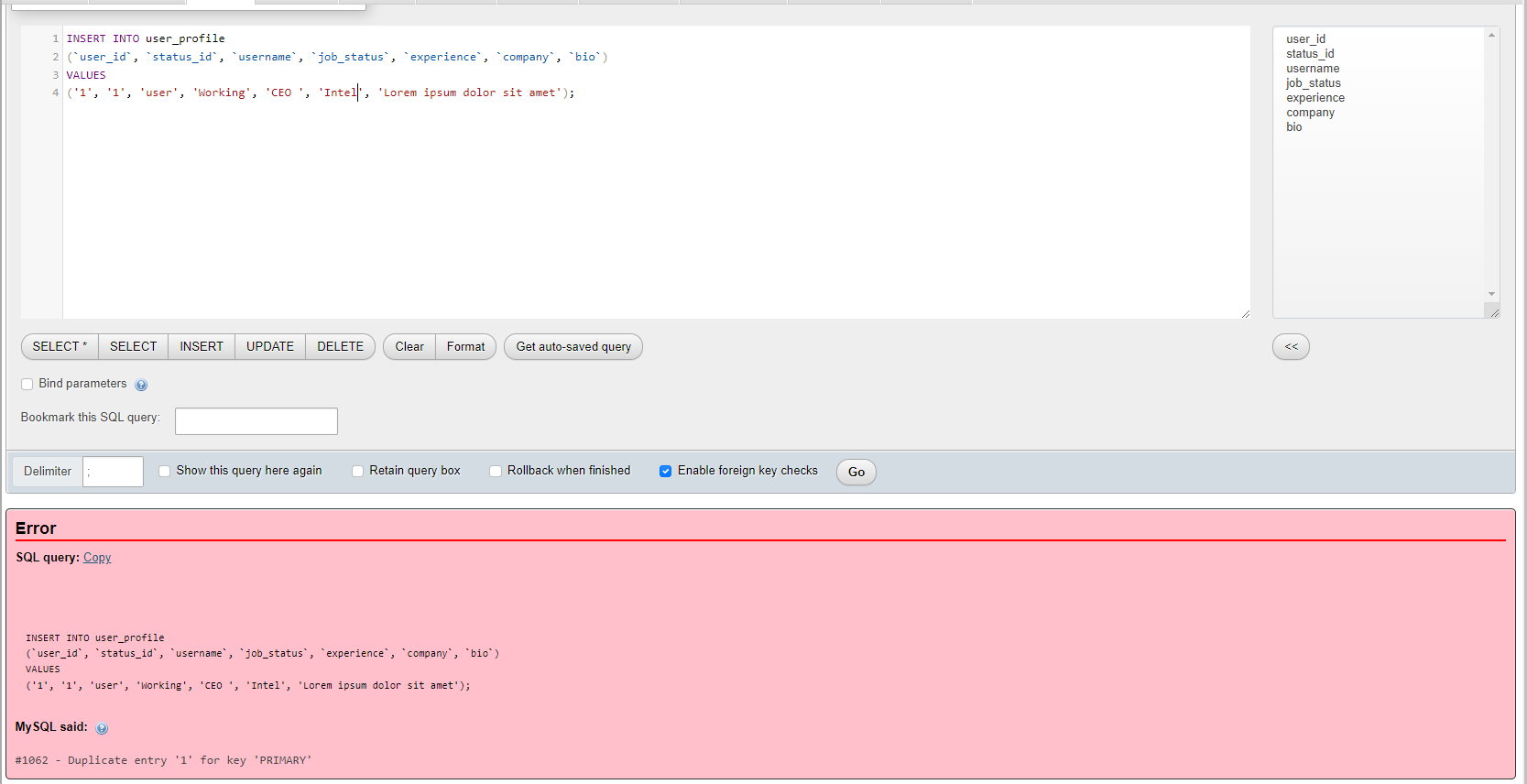
Every transaction or sequence of operations performed using SQL statement must conform to the ACID properties validation..

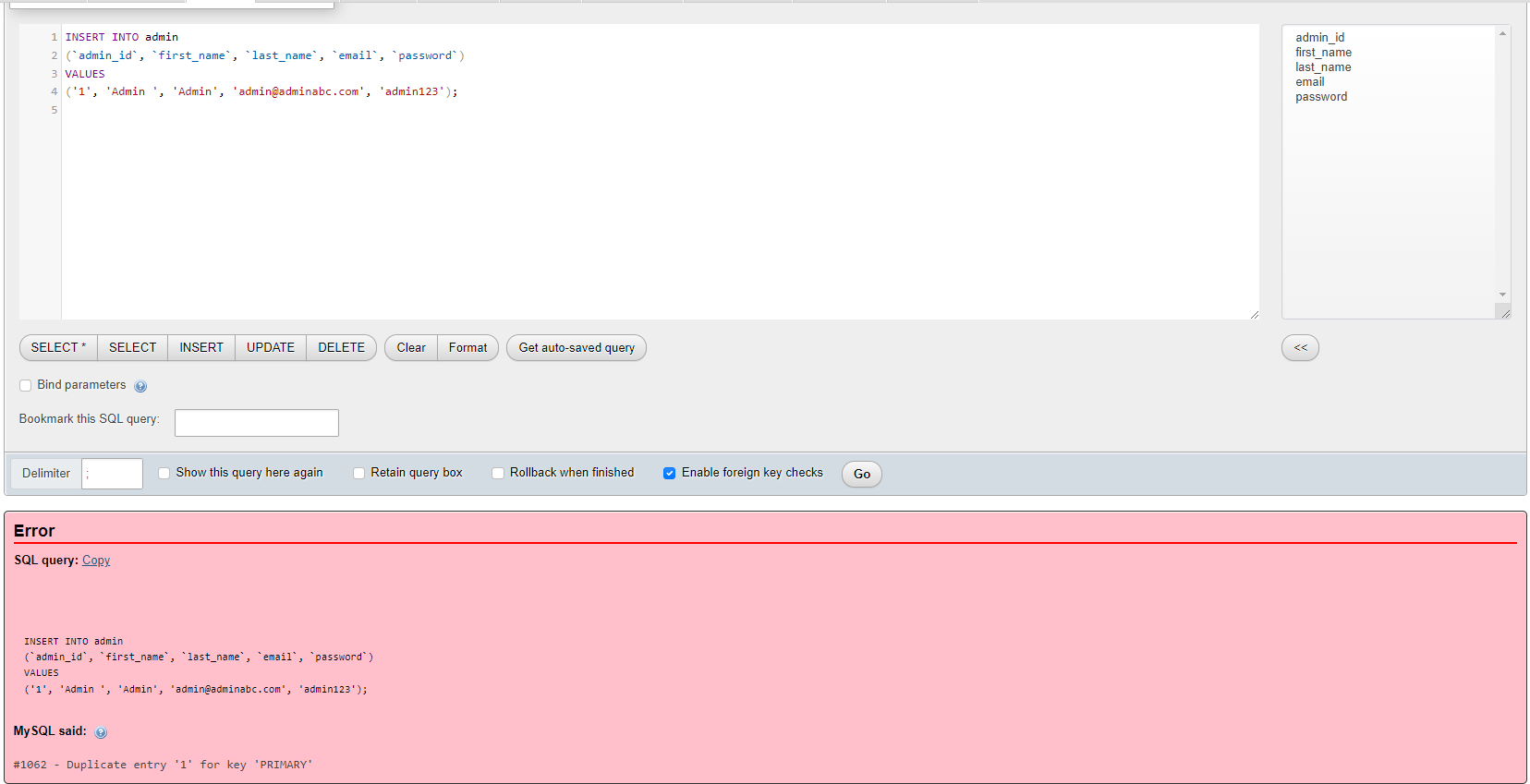
**Table/columns testing**

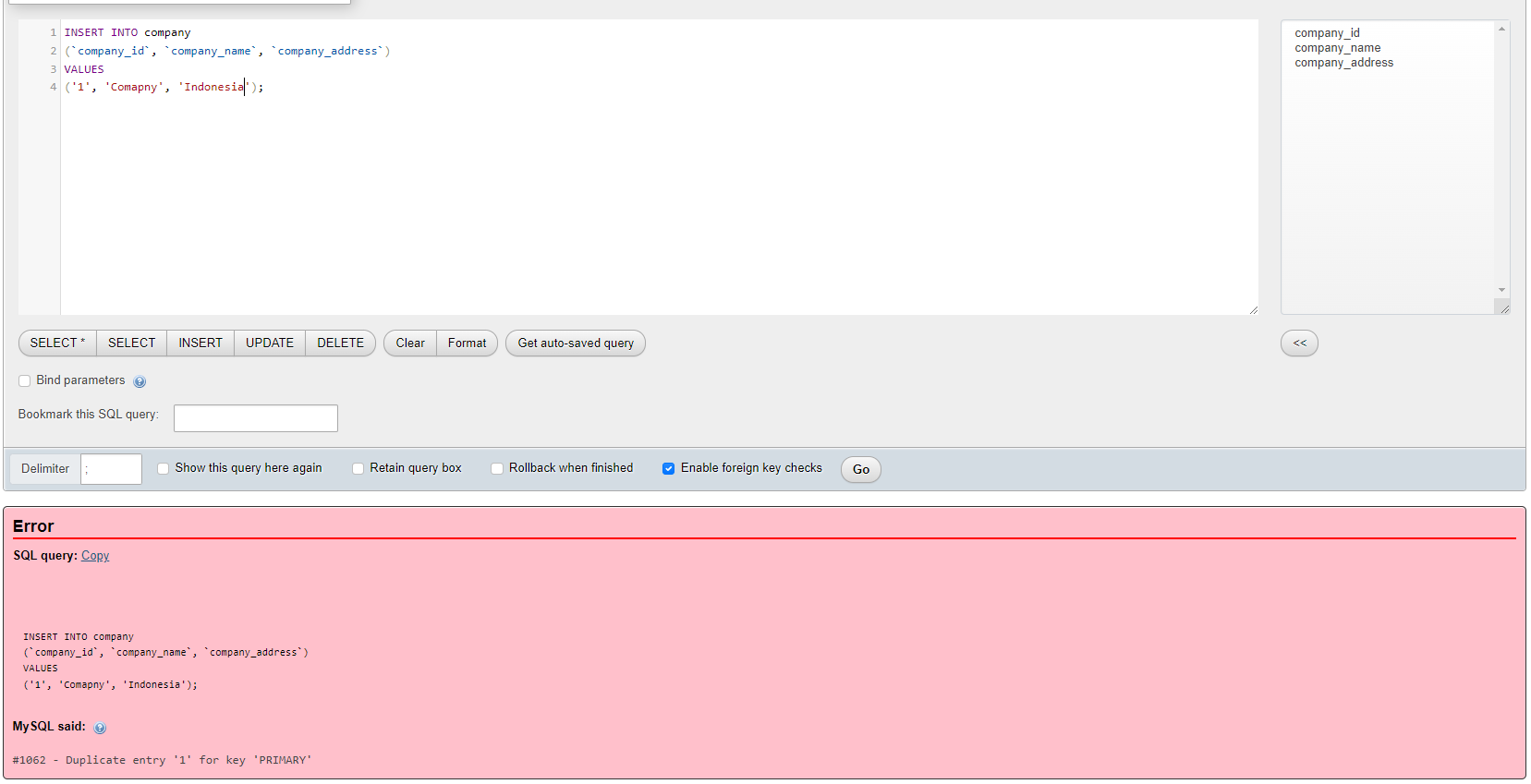
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TestCase ID | Test case | Expected result | Actual result | Pass / Fail | Evidence |
| TC01 | 1. Primary Key testing: Insert duplicate value for primary key column for ‘user’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC01 |
| TC02 | 2. Primary Key testing: Insert duplicate value for primary key column for ‘user\_profile’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC02 |
| TC03 | 3. Primary Key testing: Insert duplicate value for primary key column for ‘admin’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC03 |
| TC04 | 4. Primary Key testing: Insert duplicate value for primary key column for ‘company’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC04 |
| TC05 | 5. Primary Key testing: Insert duplicate value for primary key column for ‘job’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC05 |
| TC06 | 6. Primary Key testing: Insert duplicate value for primary key column for ‘thread’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC06 |
| TC07 | 7. Primary Key testing: Insert duplicate value for primary key column for ‘message’ table | Error message should appear because duplicate value will be inserted into primary key column violating the rule | Error message appear because duplicate value is inserted into primary key column | Pass | Evidence screen capture No TC07 |

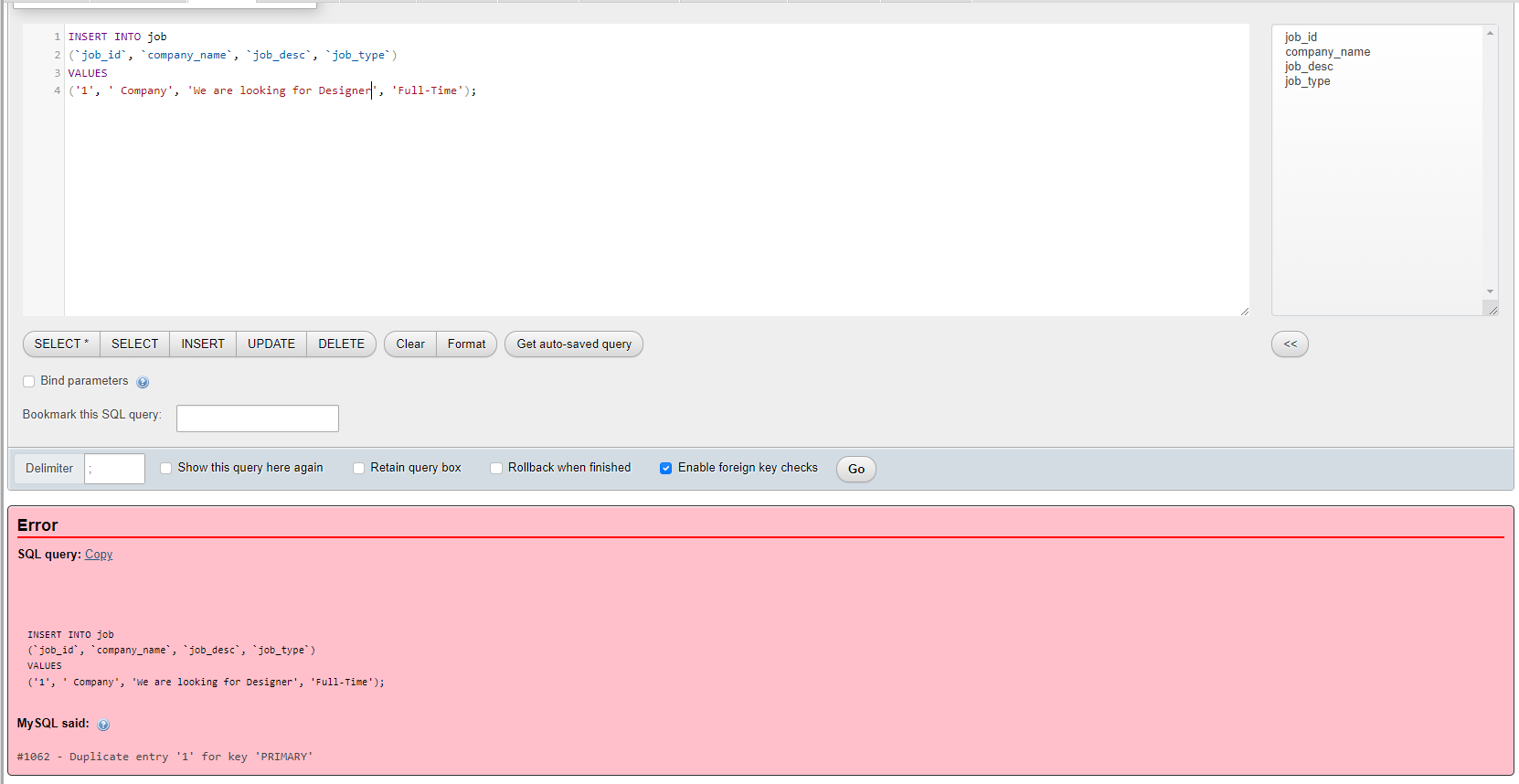
**Evidence:**

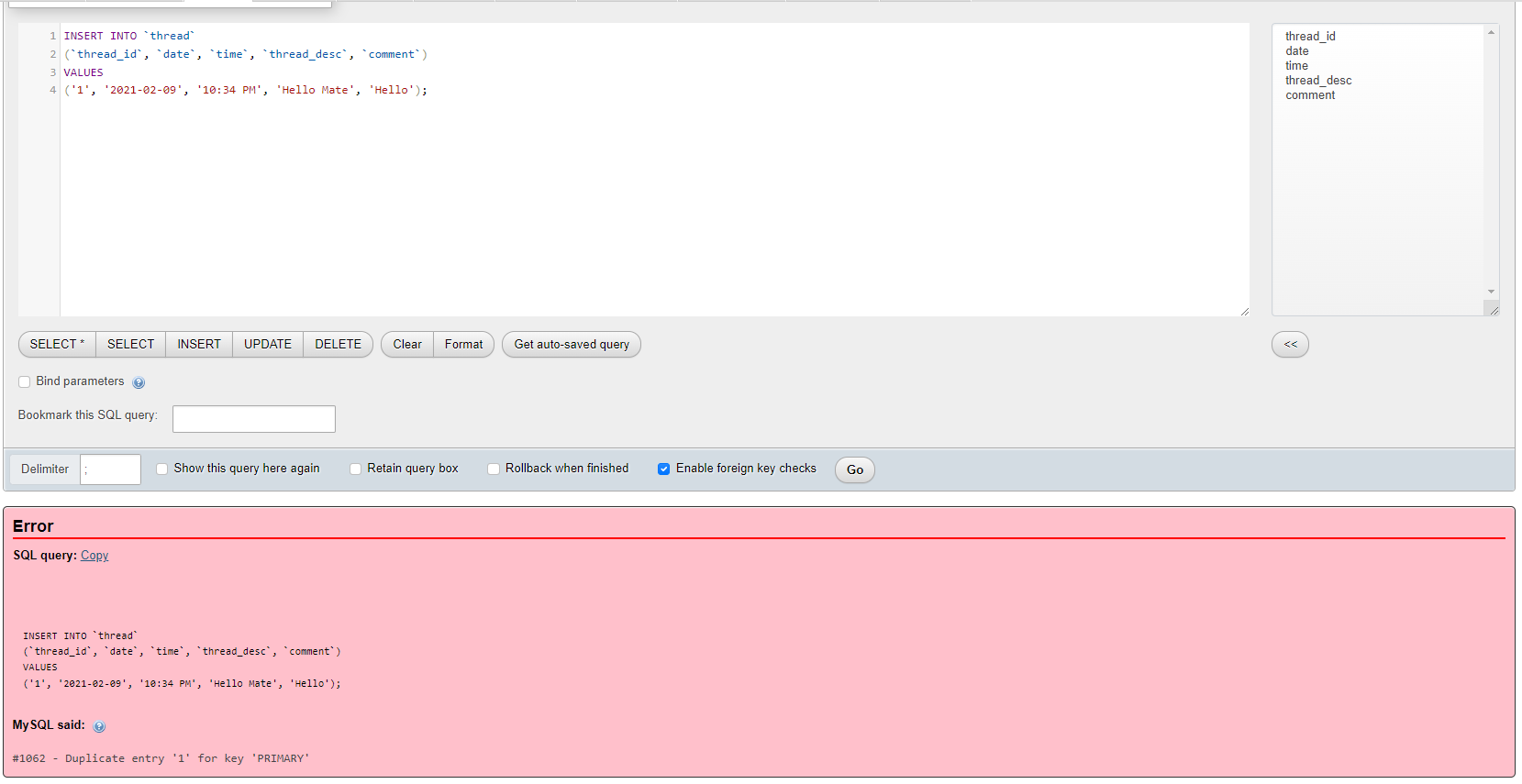
**TC01  
**

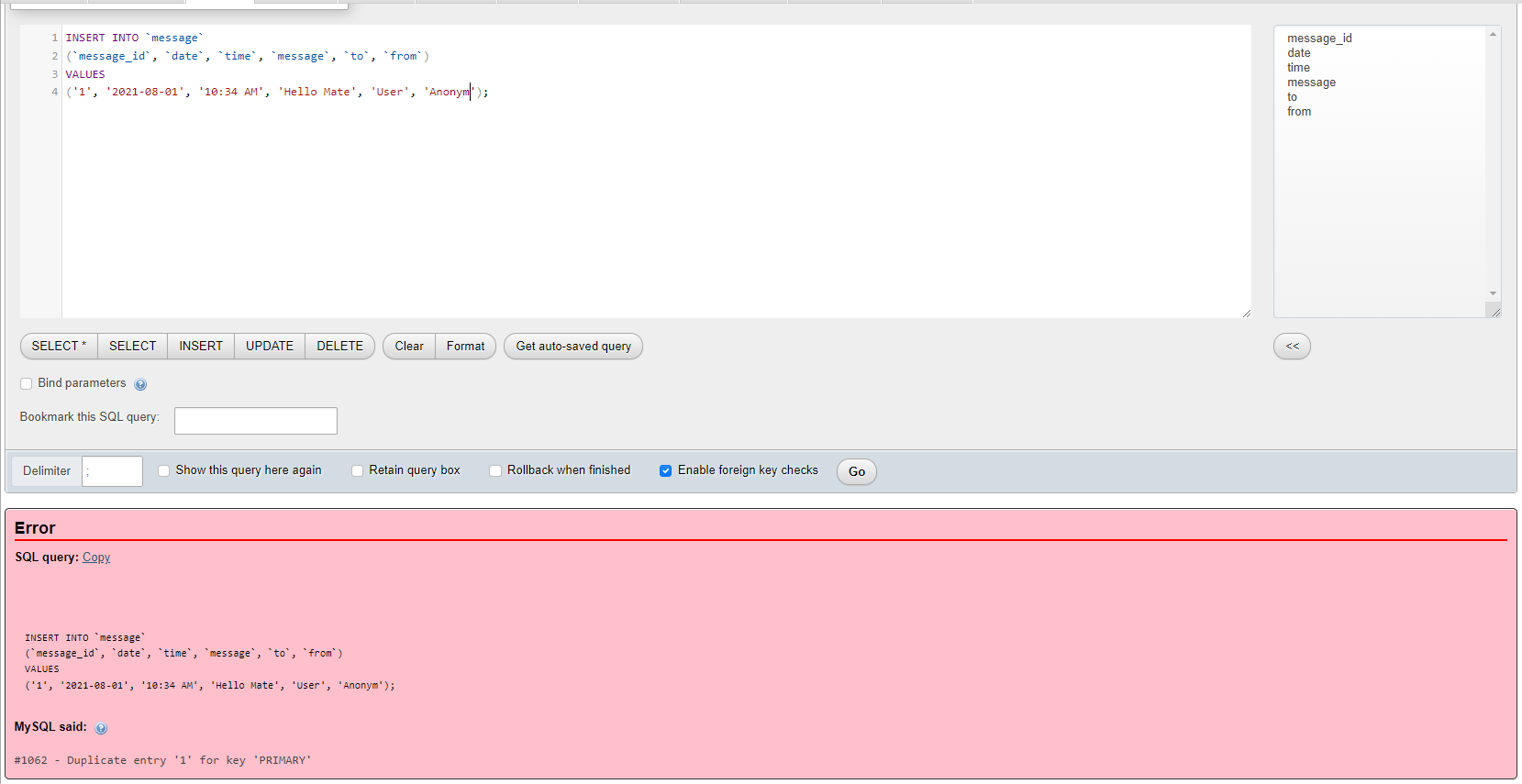
**TC02**

**TC03**  


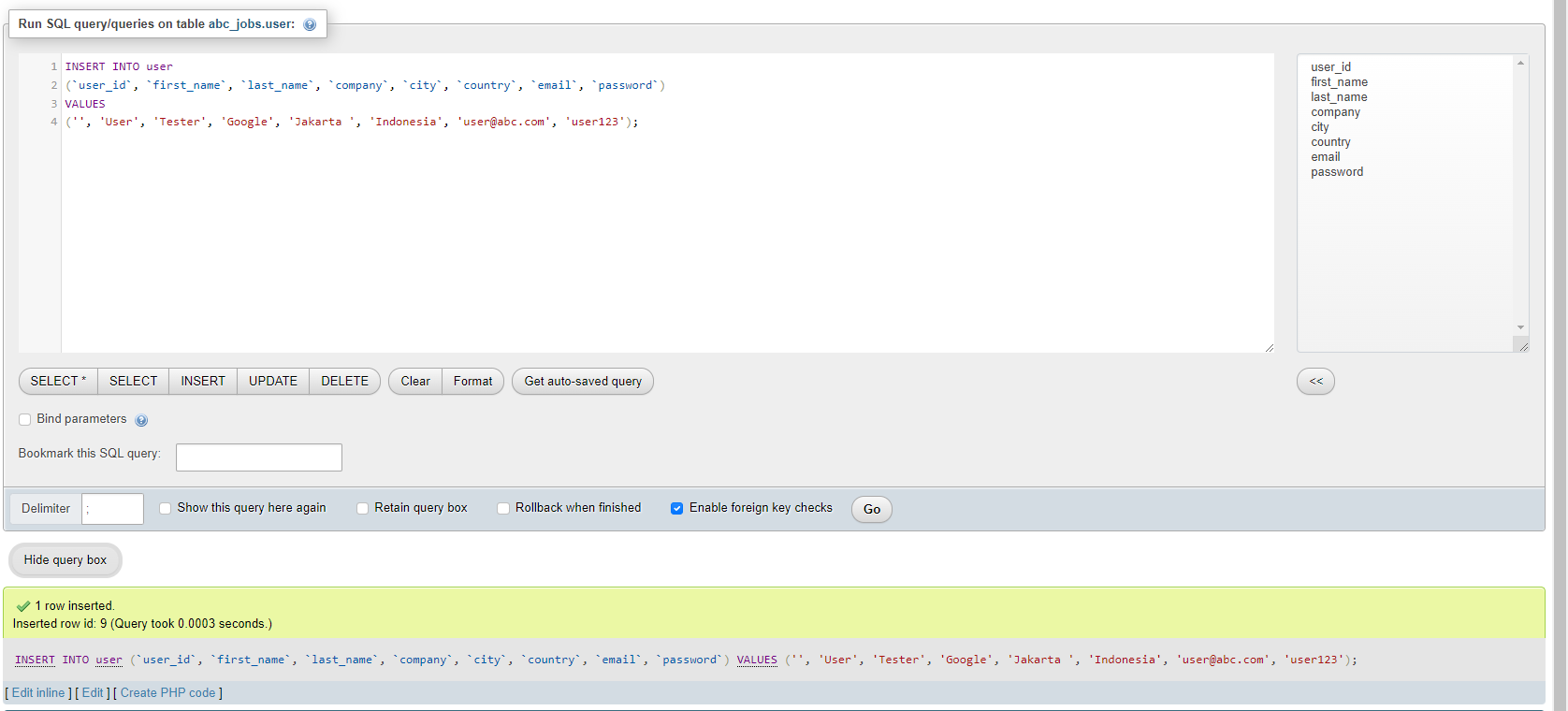
**TC04  
**

**TC05  
**

**TC06  
**

**TC07  
**

SQL Query below shows that the next increment value is 9. User, user\_id is auto incrementing.

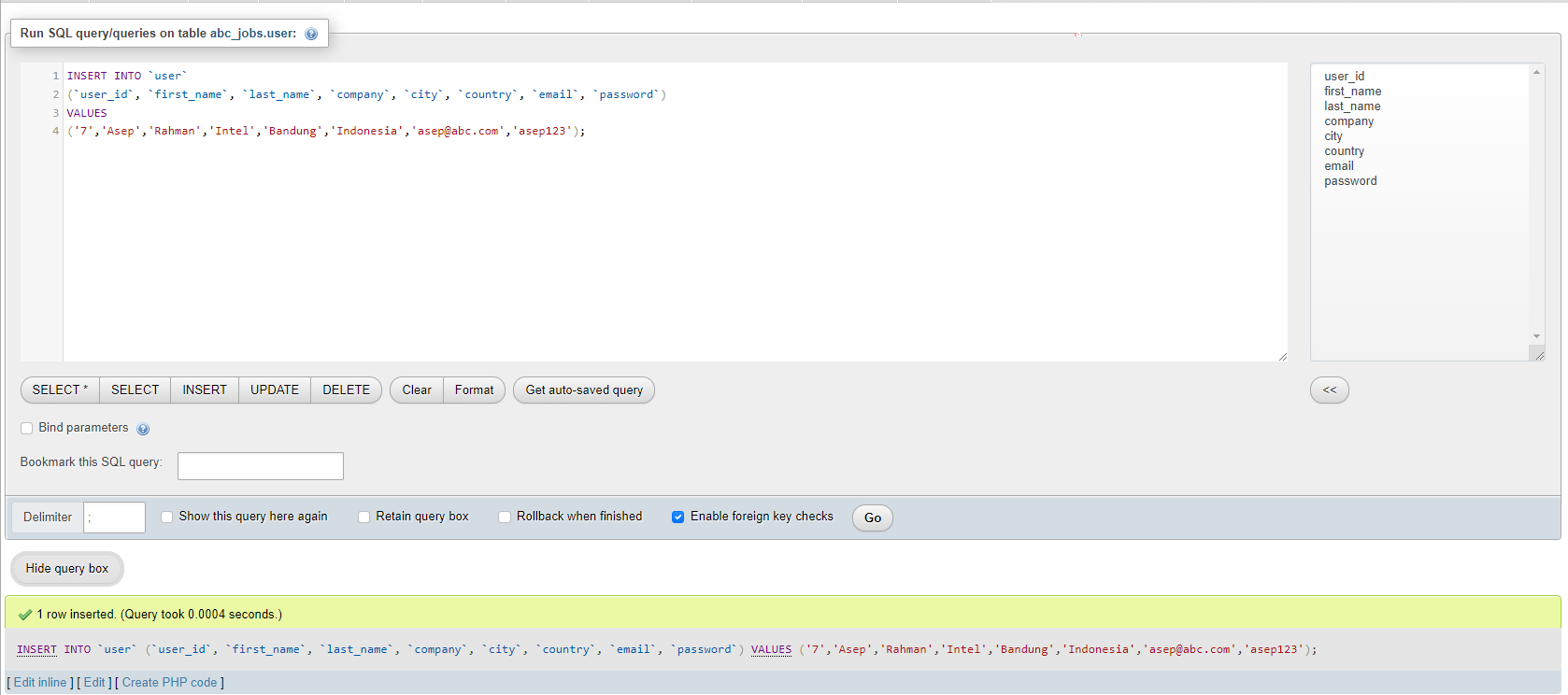


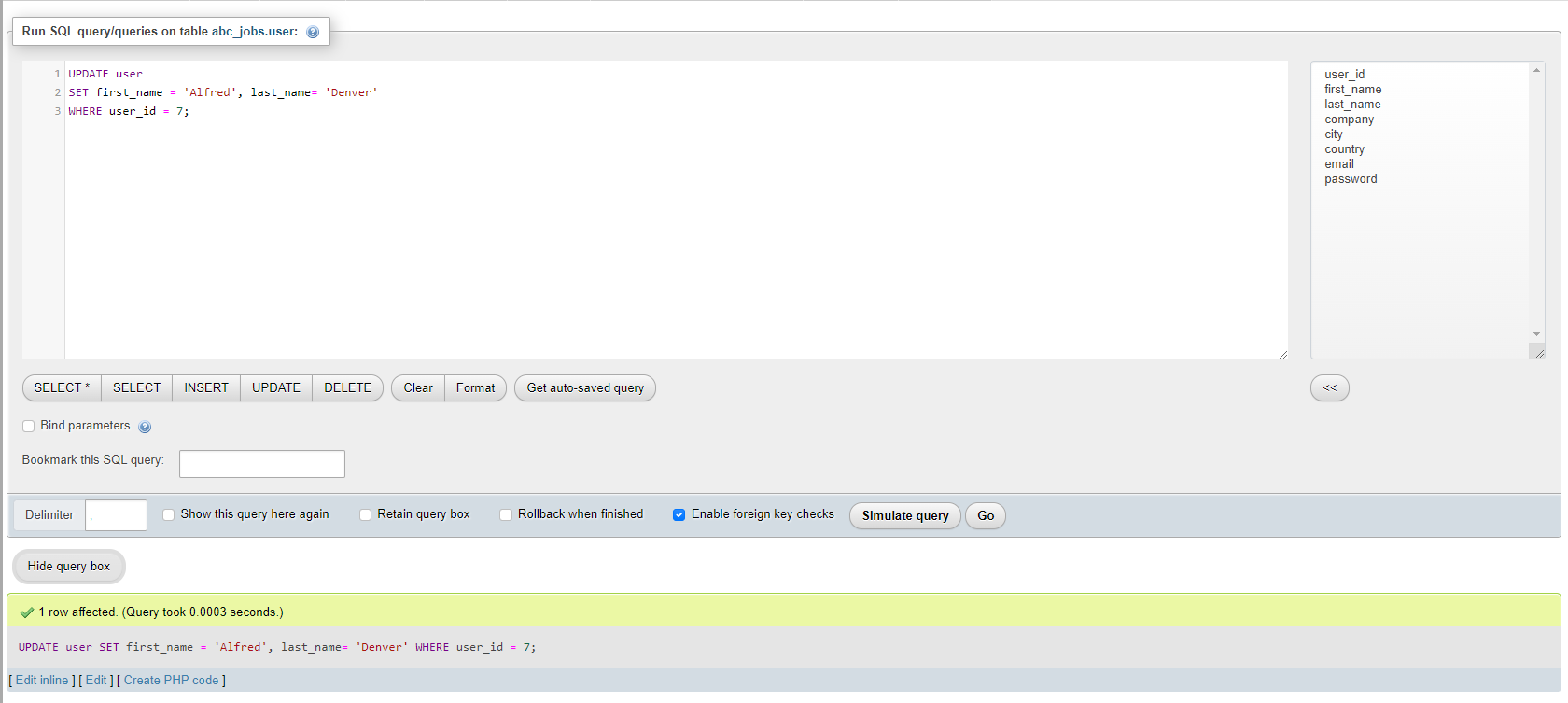
1. **Functional Database Testing**

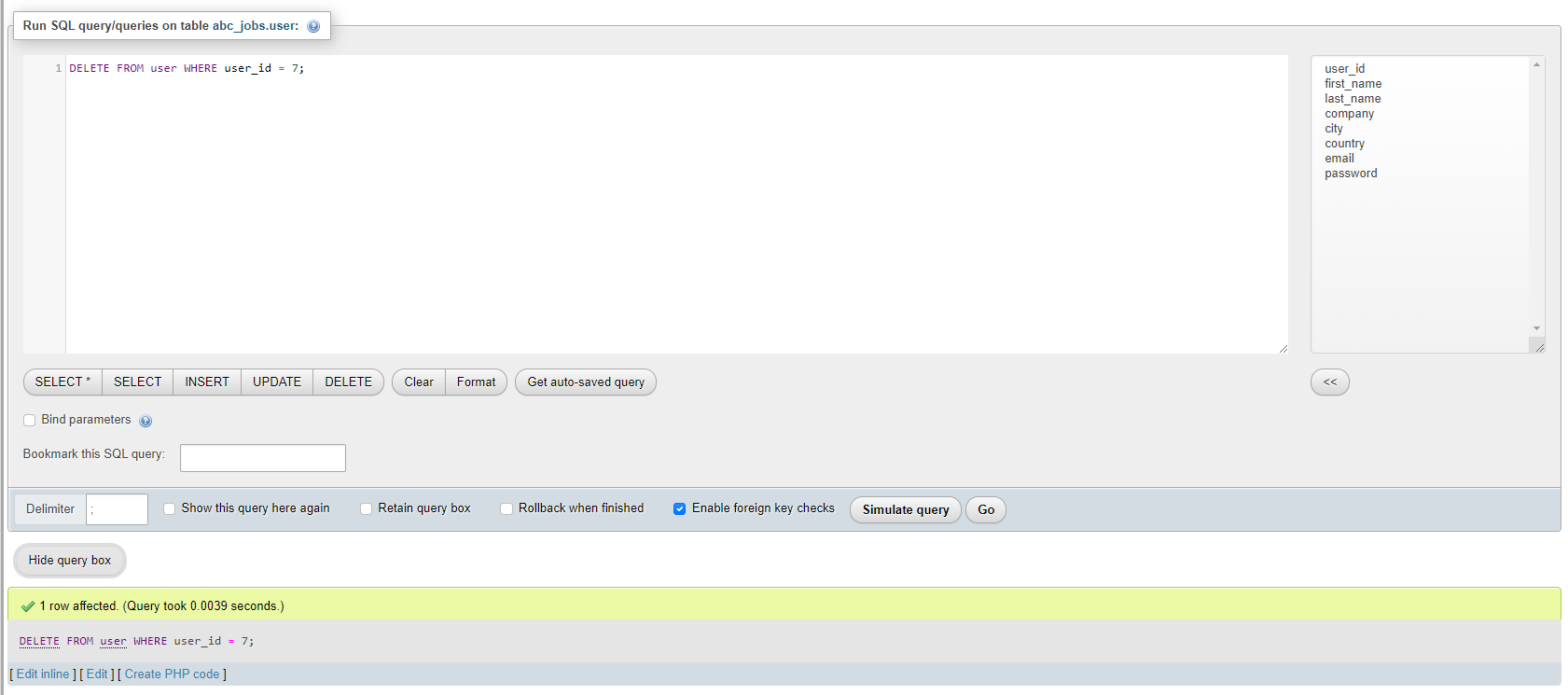
Functional database testing is to test whether data in the database can be access and updated by the users and applications. all CRUD

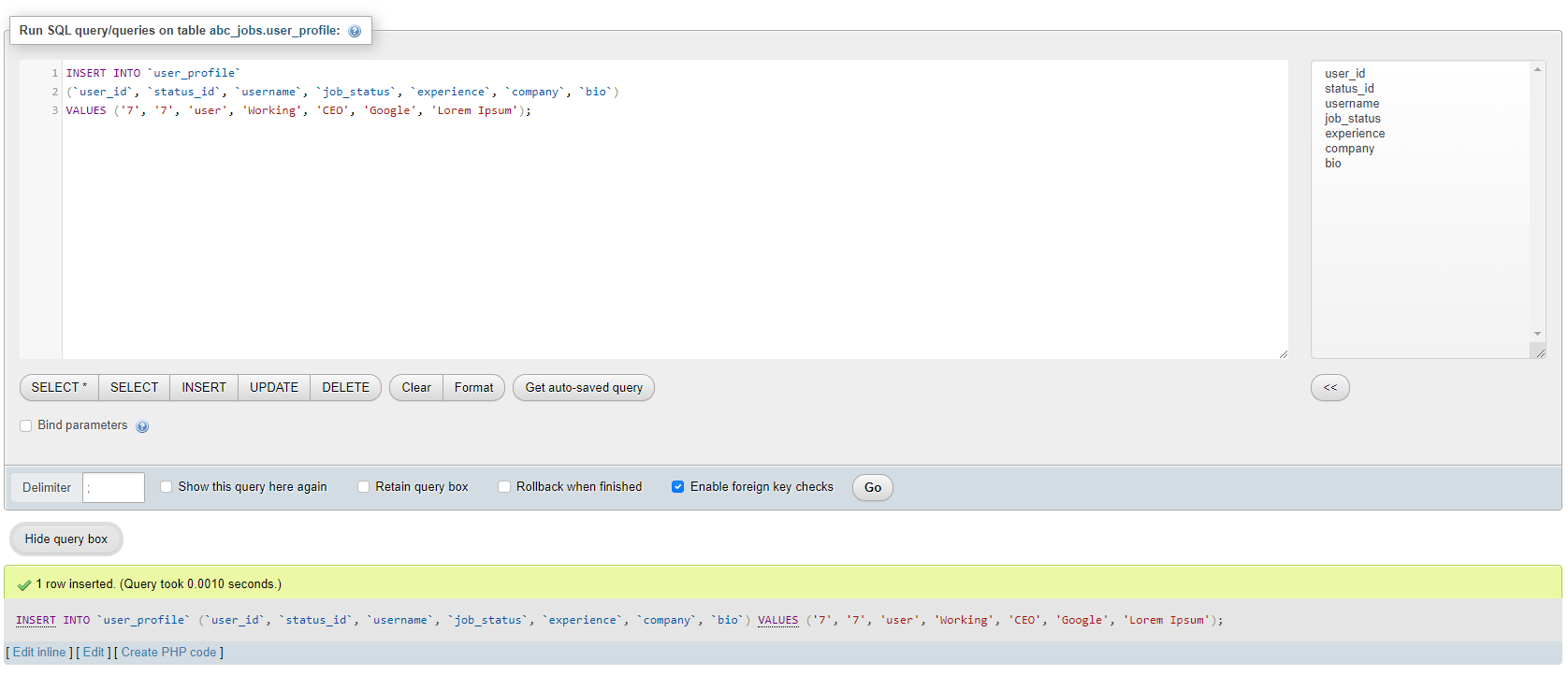
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TestCase ID | Test Case | Expected Result | Actual Result | Test Result | Evidence |
| FC001 | 1. Insert record into ‘user’ table | A record should successfully insert into 'user' table. | Record is successfully inserted into ‘user’ table | Pass | FC001 Fig |
| FC002 | 2. update record ‘user’ table | A record should successfully update 'user' table. | Record is successfully updated ‘user’ table | Pass | FC002 Fig |
| FC003 | 3. delete record ‘user’ table | A record should successfully delete 'user' table. | Record is successfully deleted ‘user’ table | Pass | FC003 Fig |
| FC004 | 4. Insert record into ‘user\_profile’ table | A record should successfully insert into 'user\_profile' table. | Record is successfully inserted into ‘user\_profile’ table | Pass | FC004 Fig |
| FC005 | 5. update record ‘user\_profile’ table | A record should successfully update 'user\_profile' table. | Record is successfully updated ‘user\_profile’ table | Pass | FC005 Fig |
| FC006 | 3. select record ‘user\_profile’ table | A record should successfully select 'user' table. | Record is successfully selected ‘user’ table | Pass | FC006 Fig |

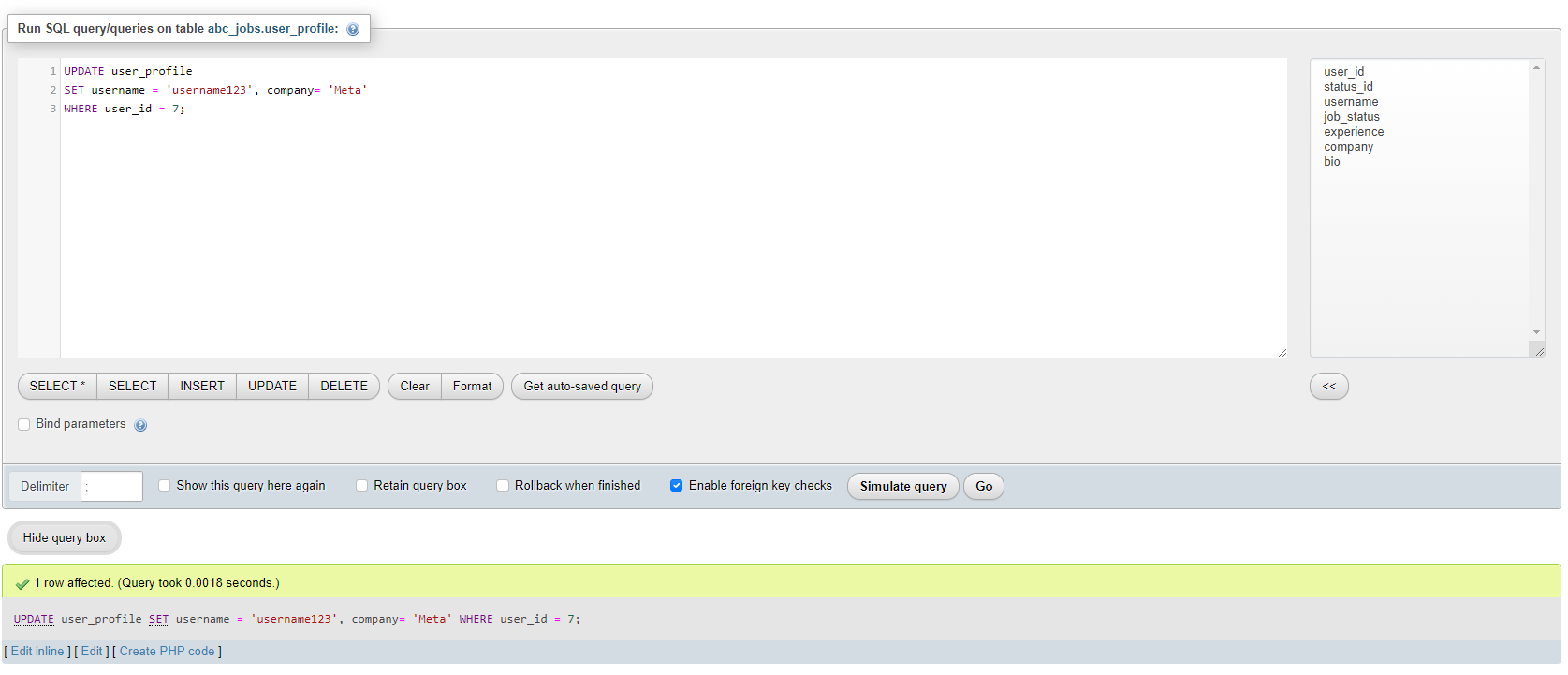
**Evidence:**

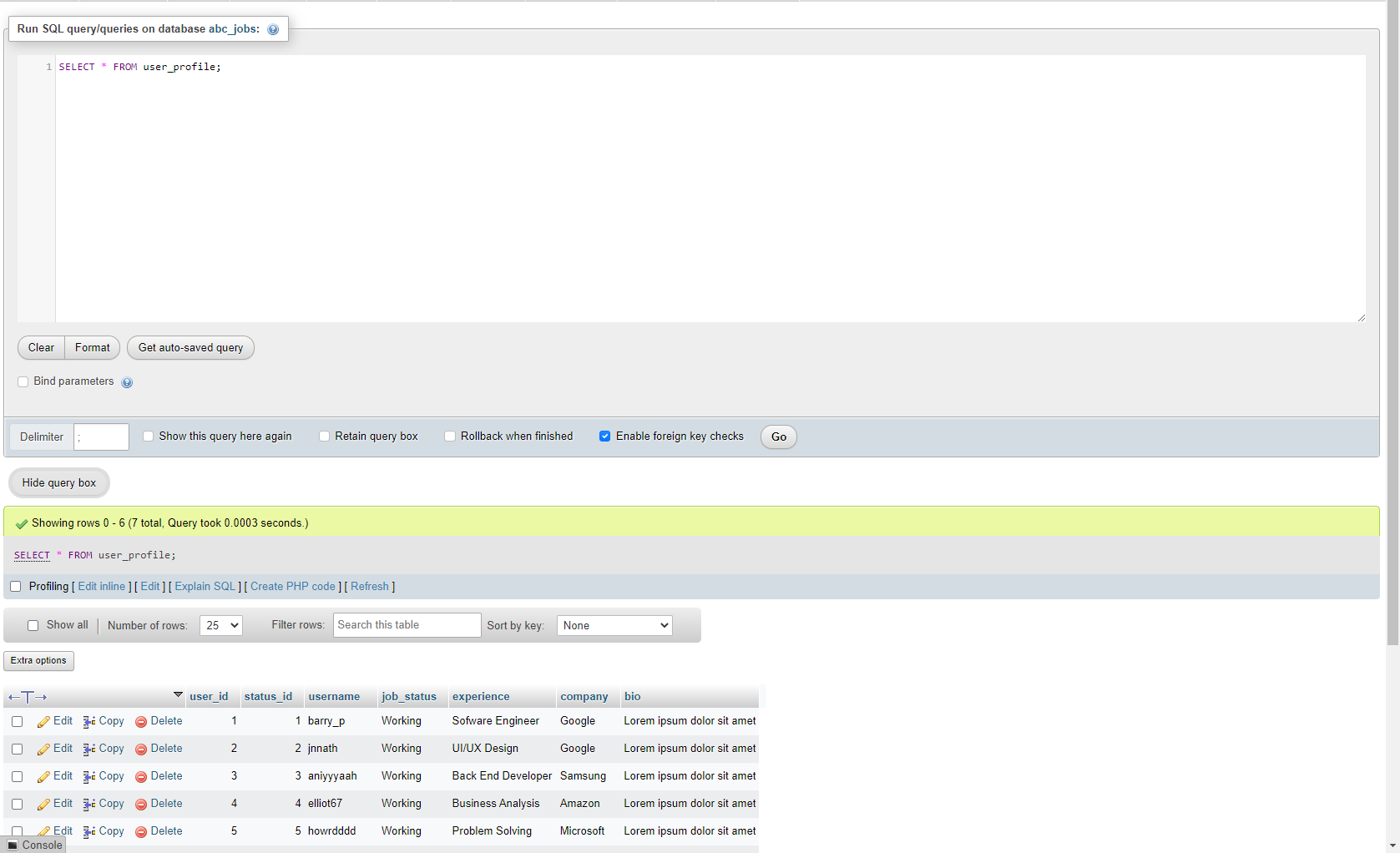
**FC001  
**

**FC002  
**

**FC003  
**

**FC004**  


**FC005**  


**FC006**

1. **Non-Functional Database**

Non – functional specify on the quality of a database characteristics or attributes based on performance, capacity, data integrity, security and more. This requirement describes more on how the product works through testing like load, stress, security and others

* 1. **Load Testing**

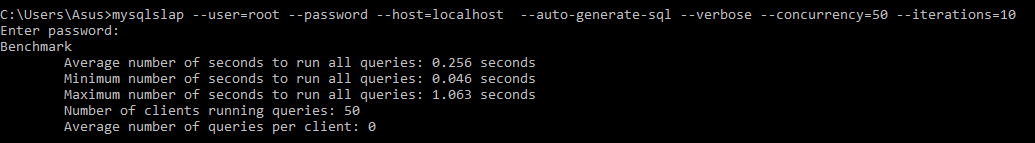
Load testing test frequently accessed transactions impact on the database performance.

This testing involves simulating real-life cases when multiple users load simultaneously (Database Testing – Performance, n.d.) .

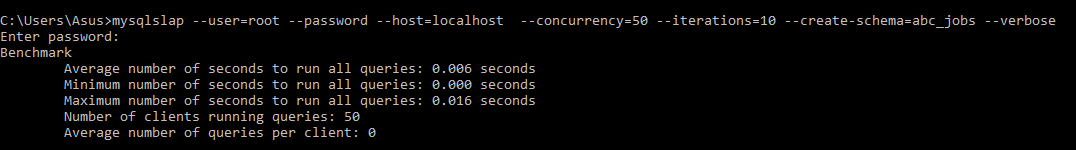
**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Test case | Note | Evidence |
| 1 | Auto generate | Test performance of the device for simple test using auto generate from the mysqlslap with 50 users and 10 repetitions. | Load test 1 |
| 2 | custom query from the auto generates | Test performance with custom query using schema from auto generate with 50 users and 10 repetitions. | Load test 2 |
| 3 | Custom test using copy of the abc database | Test performance with custom query using copy schema abc with 50 users and 30 repetitions. | Load test 3 |

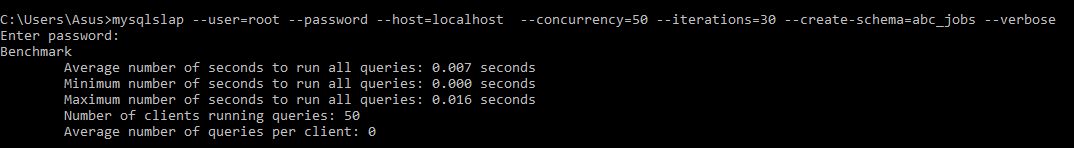
**Load test 1**



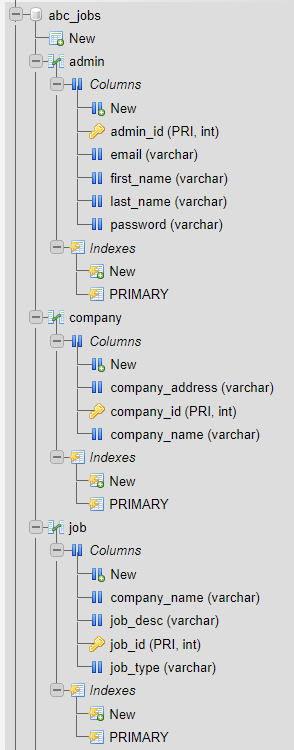
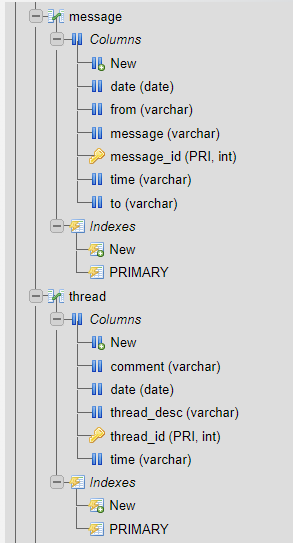
**Load test 2**

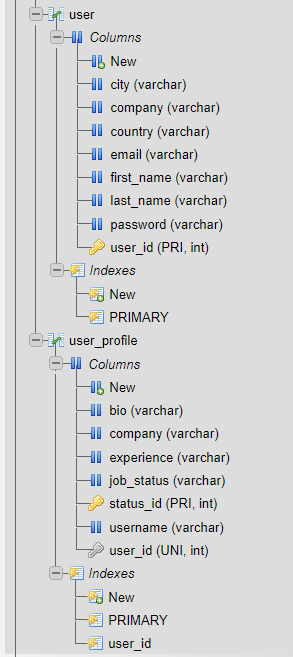


**Load test 3**



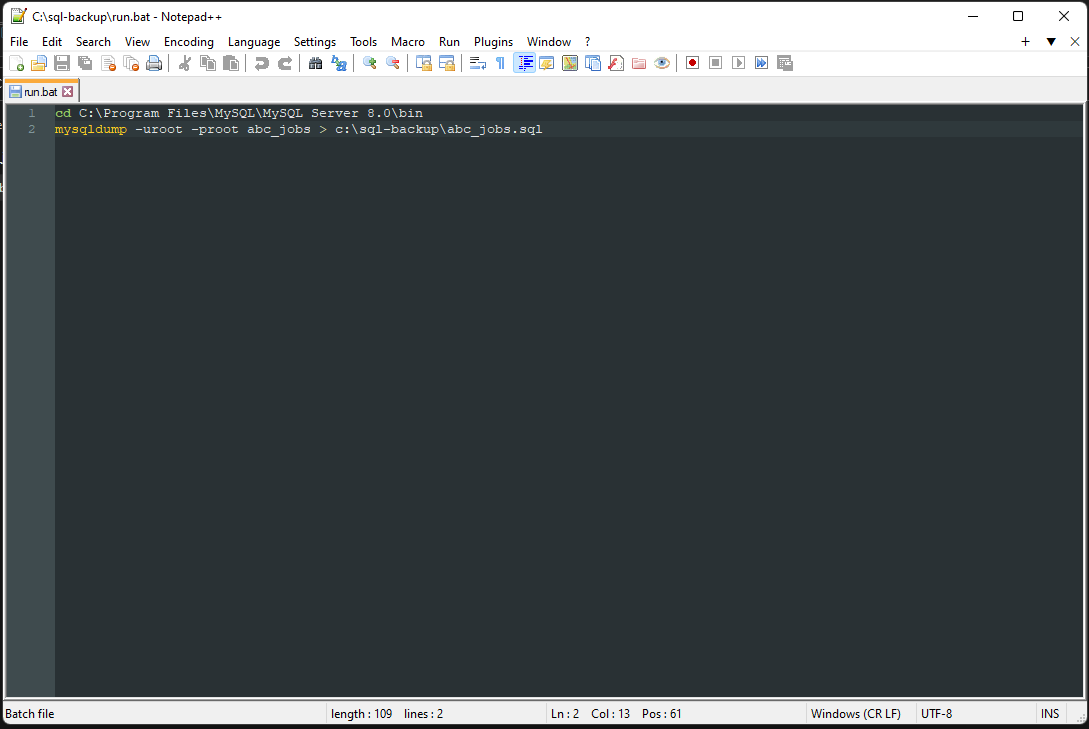
1. **Document Database**

**abc\_jobs Database   
** ****

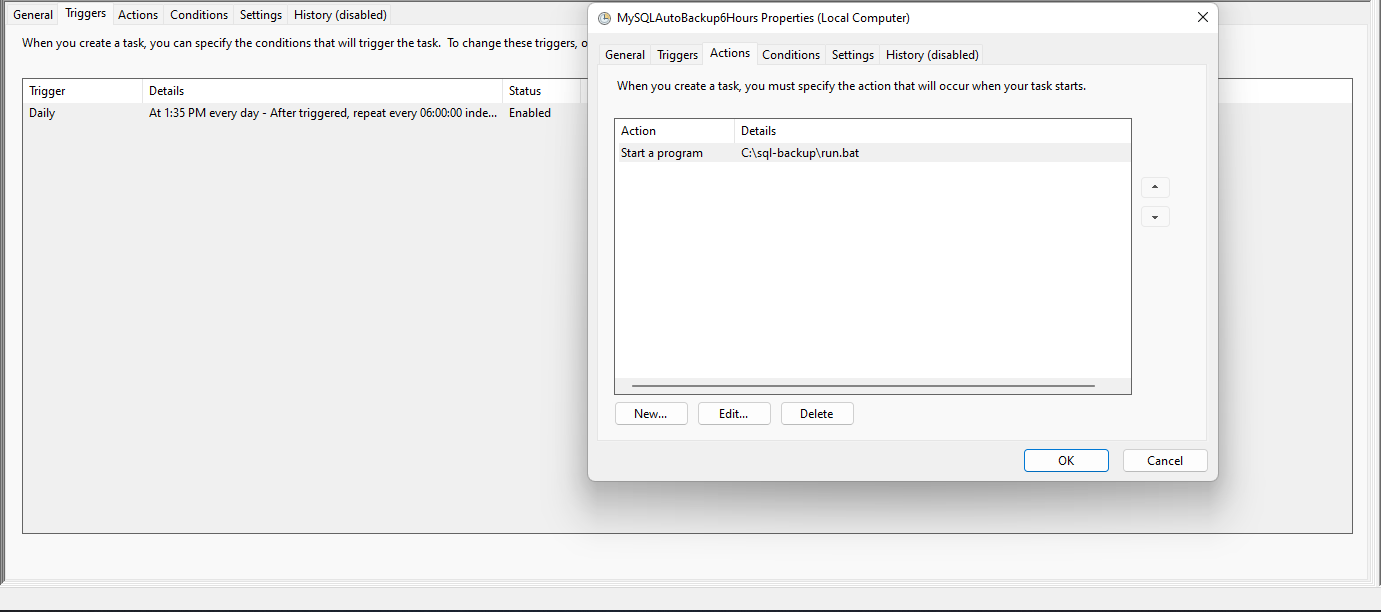
****

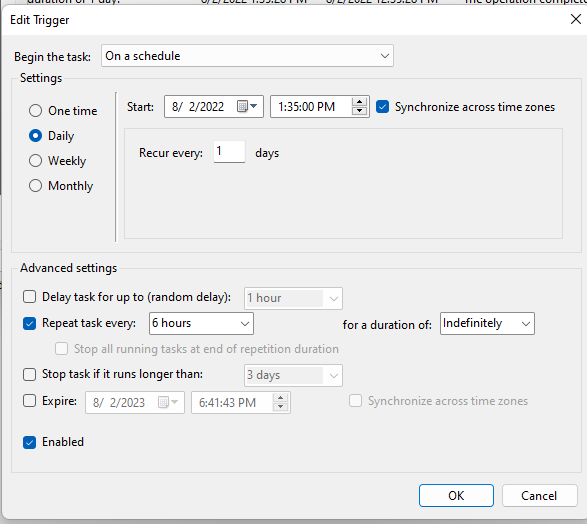
1. **Create a batch script to backup database & schedule it to run every 6 hours using windows task scheduler. Provide the script, along with the screen capture of Windows Task Scheduler**

Create run.bat file

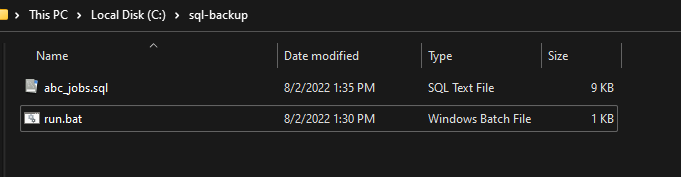


Create new task scheduler





Results



1. **Provide a Restoration script in case of failure**

****