

Linking an Infilink Project to a MySQL Database

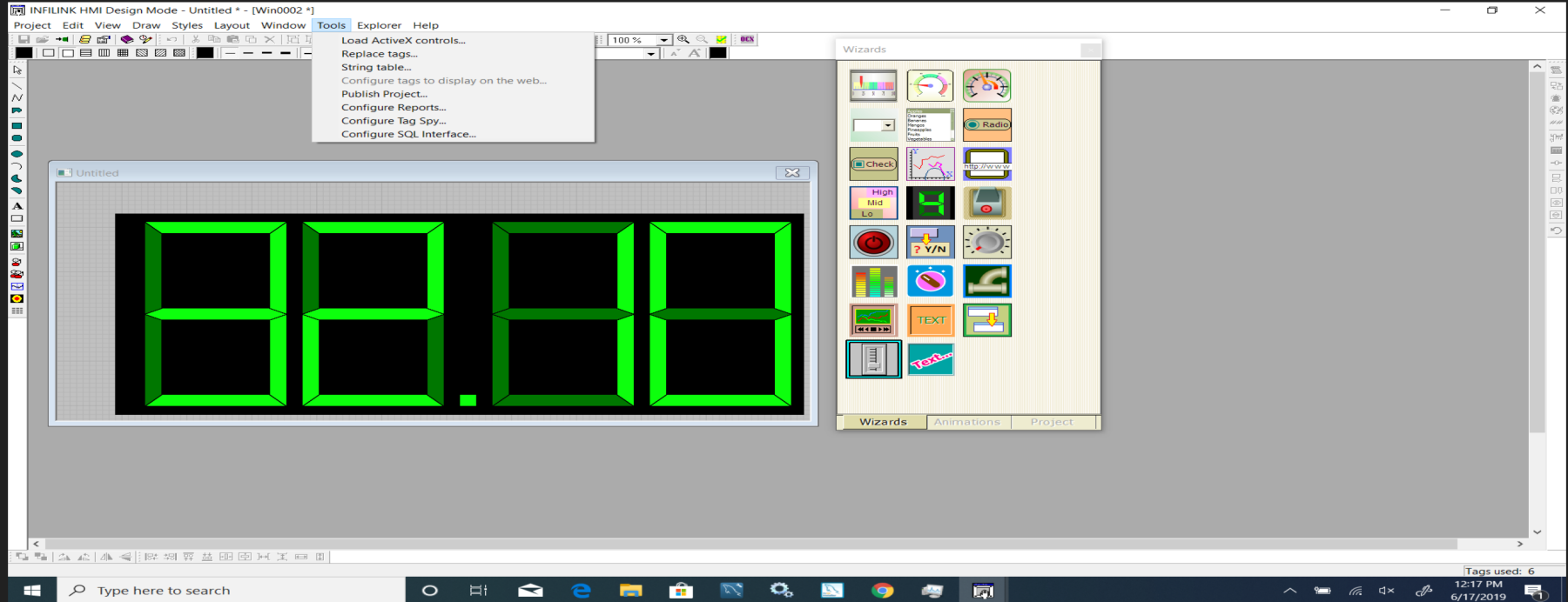


Kessler Ellis Products

by Justin Callahan

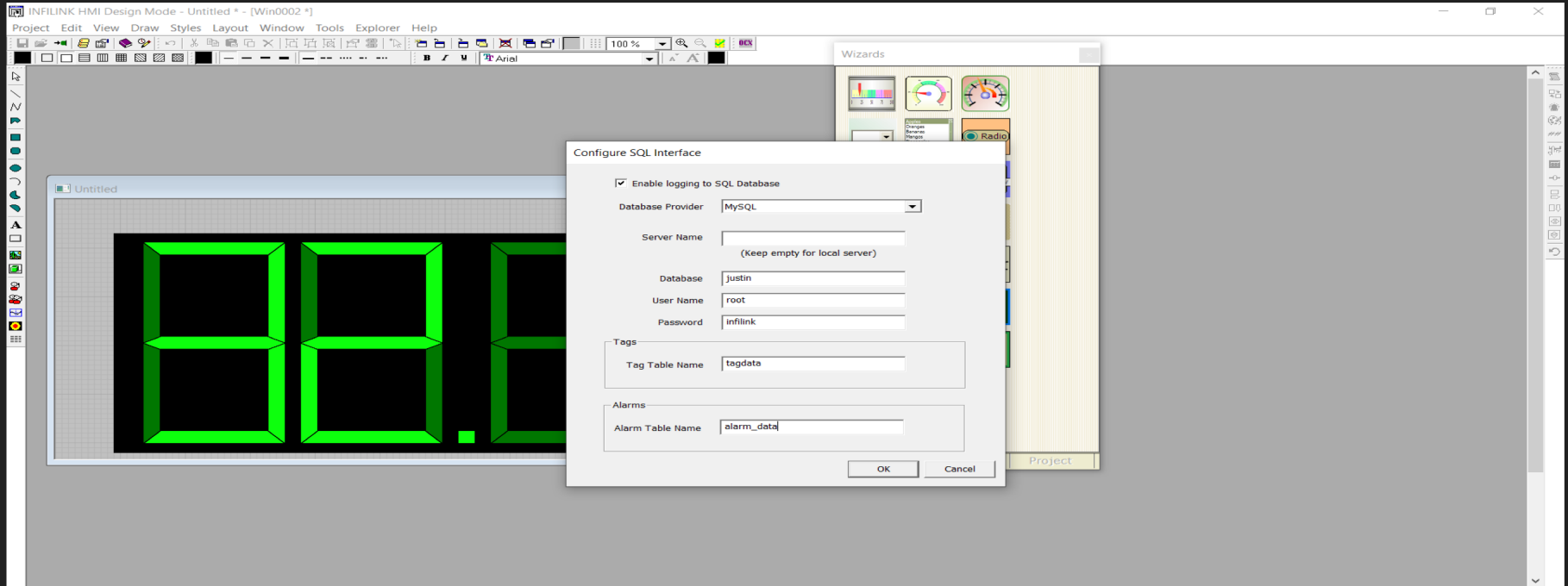
Step 1: Configure SQL interface

Under the tools drop-down menu, find configure SQL interface down at the bottom.



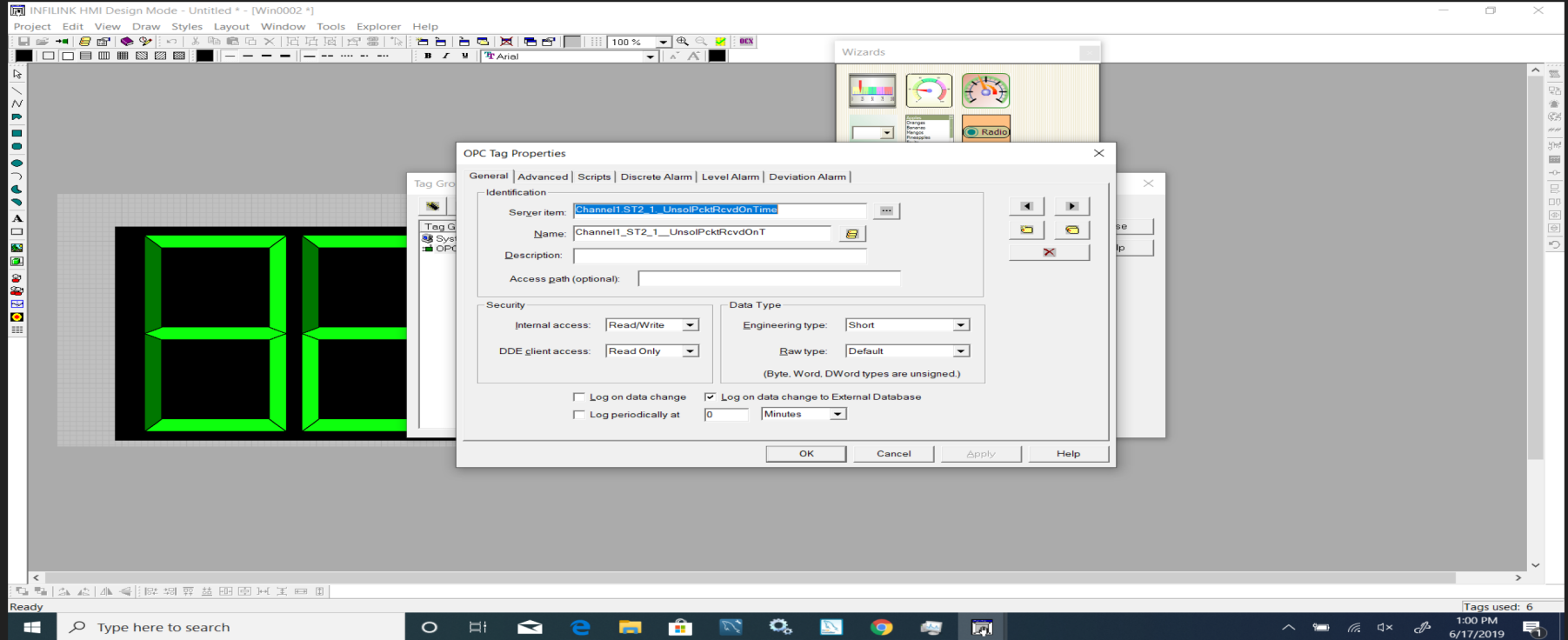
Step 2: Configuring MySQL/Infilink settings

Next, you'll want to make sure this window has your database name (you can leave it blank for now or pick the name you plan on using when you create the database). You also want to input "root" as your username and the password for root that you set up when you installed MySQL. Finally, you can pick the names of the tables that you plan on using for your tags and alarms.



Step 3: Making sure the tags are accessible outside of Infilink

You should open up the Tag Properties window and check the box labeled “Log on data change to External Database”.



Step 4: Internet

You're going to want to check to make sure the "Enable tag value access over Internet" box is checked. Left unchecked, it may interfere with transferring data into your MySQL database later on.

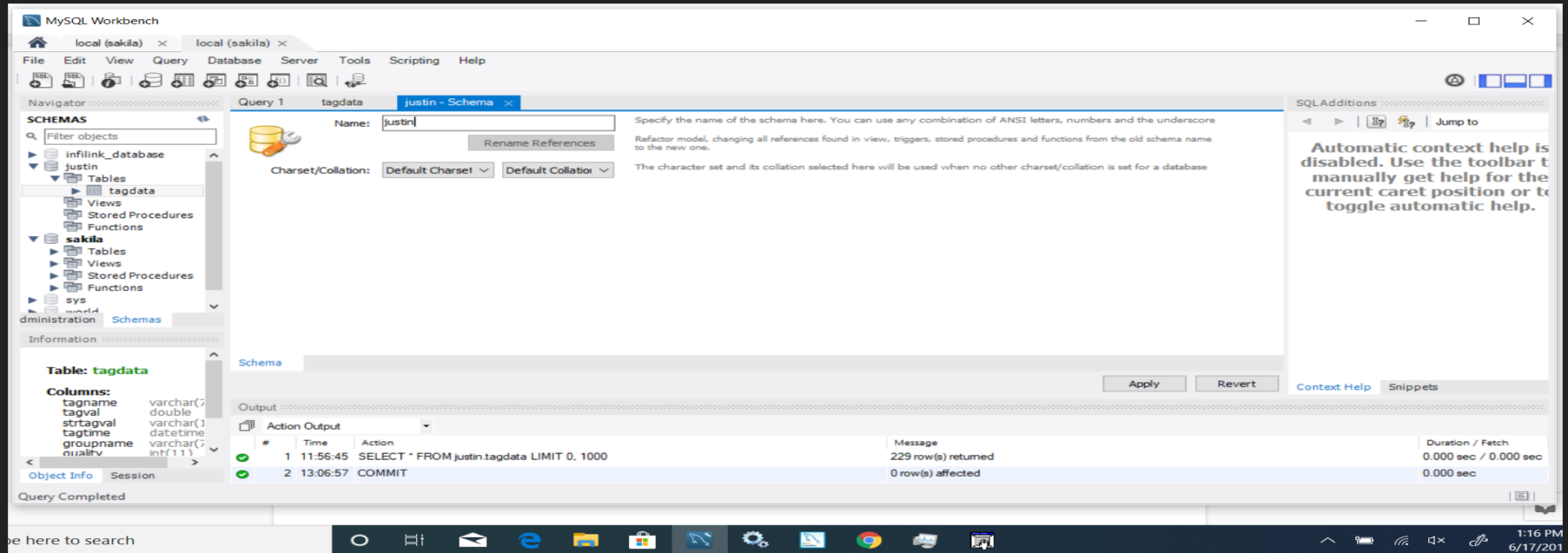
The screenshot shows the INFINLINK HMI Design Mode interface. The main workspace displays a large green digital display showing the number '88'. Overlaid on this is the 'Project Properties' dialog box, specifically the 'Internet' tab. The dialog box contains the following information:

- General** | **Password** | **User List** | **Alarm Logger** | **Data Logger** | **Scripts** | **Alarm Printer** | **Internet** | **Timer List**
- Please use menu item "Tools - Configure tags to display on the web." to select the tags, values of which are to be shown through Internet browser.
- ☒ Enable tag value access over Internet
- User details for sending emails (alarm events, SendEmail command)
 - Yahoo**
 - UserName:
 - Password:
 - Gmail**
 - UserName:
 - Password:
- ☐ Enable sending E-mail on Alarm events
 - ☒ Send using Yahoo ☐ Send using Gmail
 - Recipient's E-mail address:
- Send E-mail for alarms with severity levels
 - ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7
 - ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15
- Buttons: OK, Cancel, Apply, Help

In the bottom right corner, there is a yellow warning icon and a message: "KEPServerEX 6.6 One or more features are time limited. Server - Administration". The system tray at the bottom shows the date and time as 6/17/2019 1:11 PM.

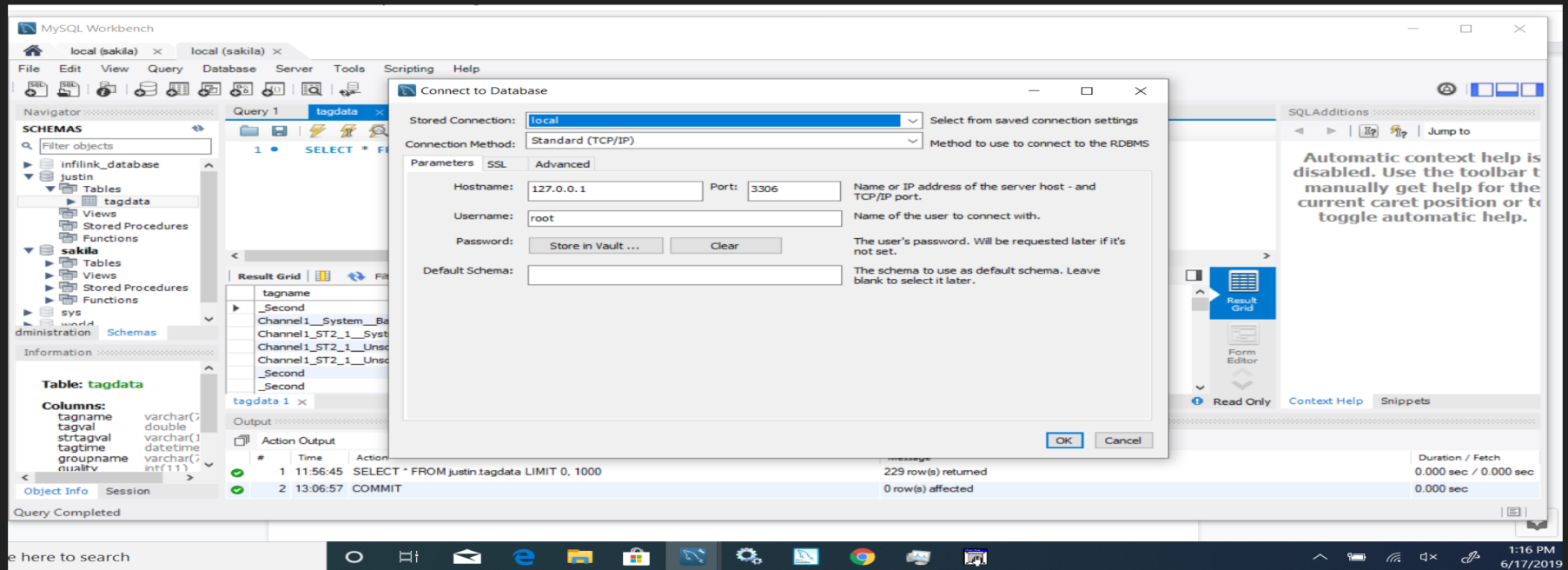
Step 5: Creating a database

To create a database, you're going to want to create a Schema from the toolbar in MySQL. Your Schema/database here must have the same name as the one in the configuration setting on Infilink.



Step 6: Connecting to your localhost

To connect to your localhost, create a database connection using your root username and password, and your own IP address or “localhost” as the hostname.



Step 7: Creating your tables

Within your newly created database, create two tables, one for tag information with the columns above, and one with alarm information with the columns above (These should have the same respective names as the tag and alarm tables in the configuration window in Infilink).

Query 1 tagdata tagdata tagdata - Table **tagdata - Table**

Table Name: tagdata Schema: **inflink_database**

Collation: utf8 - default collation Engine: InnoDB

Comments: Inflink Tag Data

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defi
◊ tagname	VARCHAR(70)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ tagval	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ strttagval	VARCHAR(128)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ tagtime	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ groupname	VARCHAR(70)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defi
◊ tagtime	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ groupname	VARCHAR(70)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ quality	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL

inflink_database

Tables

alarm_data

tagdata

Table Name: alarm_data Schema: **inflink_database**

Collation: utf8 - default collation Engine: InnoDB

Comments: Inflink alarm table

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defi
◊ groupname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ tagname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ tagval	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ almtime	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ almtyp	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL

Query 1 tagdata tagdata inflink_database.alarm_data **alarm_data - Table**

Table Name: alarm_data Schema: **inflink_database**

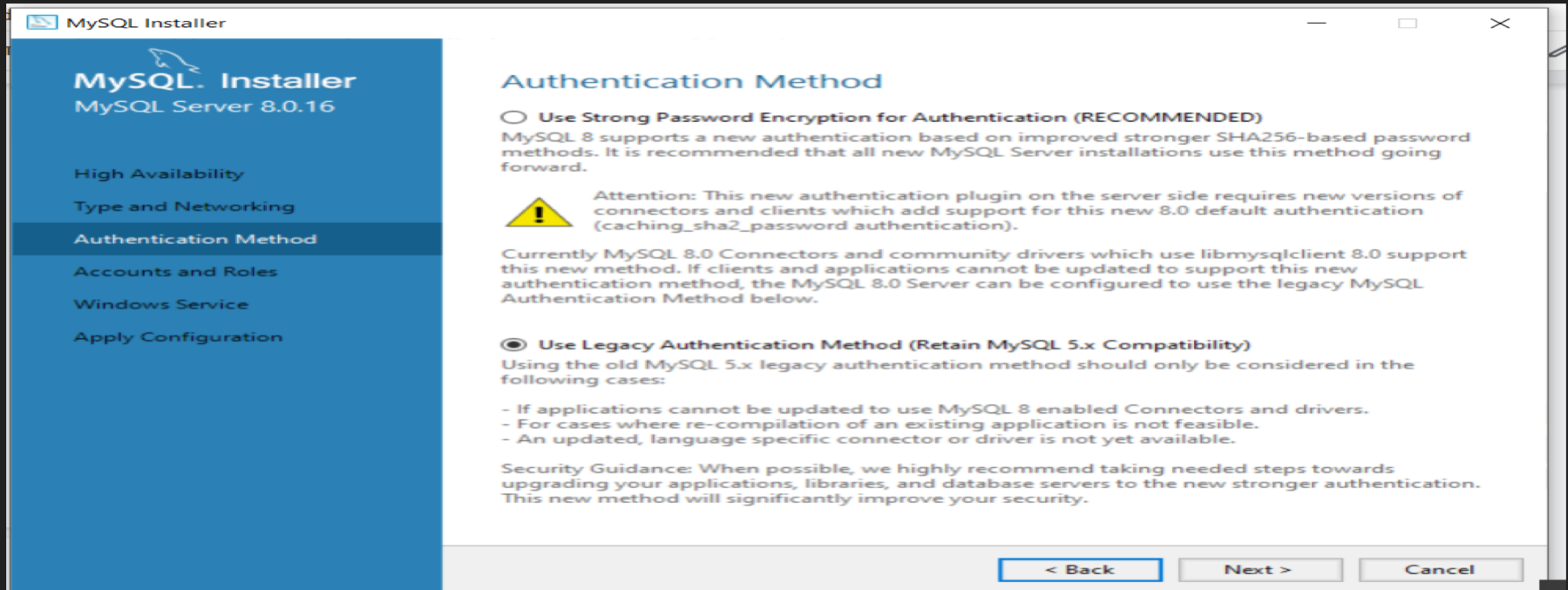
Collation: utf8 - default collation Engine: InnoDB

Comments: Inflink alarm table

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defi
◊ almtyp	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ event	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ severity	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ comment	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL
◊ user	VARCHAR(70)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NUL

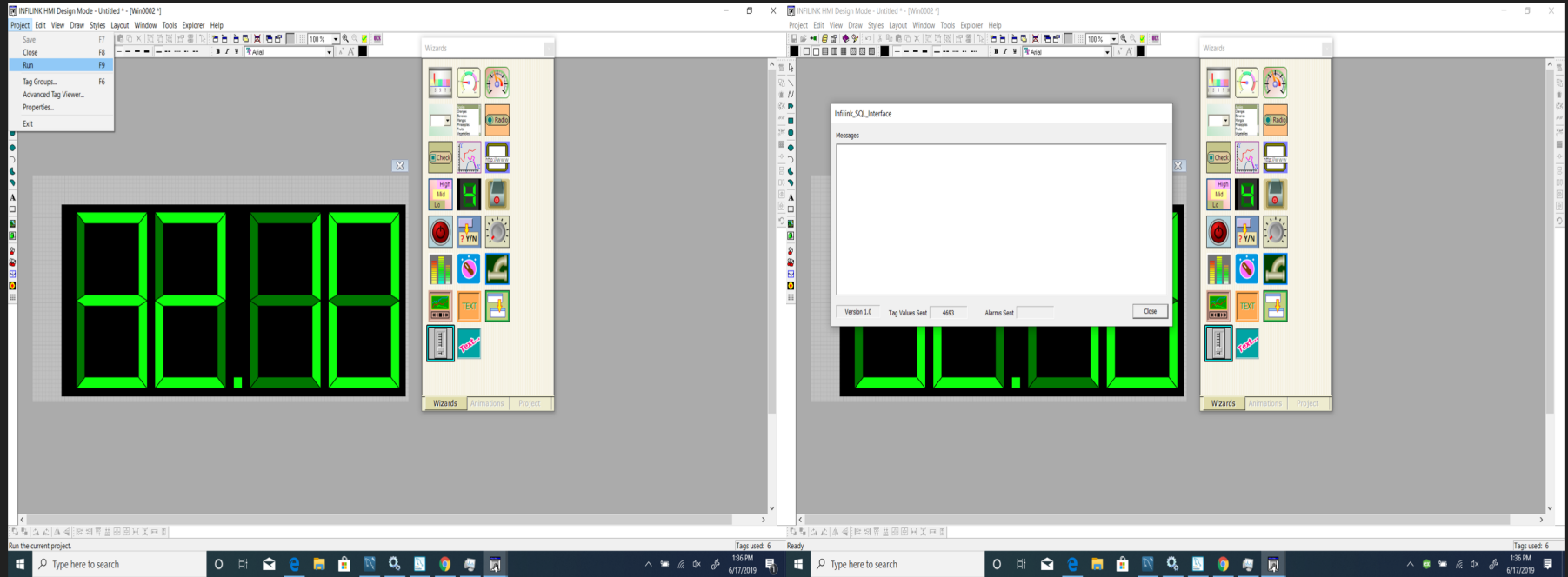
Step 7.5: Legacy Authentication

This step may be unnecessary for some of you, but for those of you who chose the “Strong Password Encryption for Authentication” option when you were installing the MySQL server, you’re going to want to go back into the installer and change that to the “Legacy Authentication” option. Otherwise you will find that MySQL and Infilink are incompatible.



Step 8: Running your Infilink project

You're going to want to run your Infilink project in order to send the data over into your MySQL database. When you run it, a screen (shown below) should pop up. If yours looks like this one, that is good news. If there is anything in the white space, that usually means that an error has occurred. You should address whatever error code appears in the white space.



Step 9: Check your tables

Once your tables are created, a line of code as shown below should appear in a terminal. If all of the previous steps have been completed successfully, then below said terminal, your table should appear with all of your respective tag data or alarm data. Congratulations, you have successfully linked Infilink and MySQL.

MySQL Workbench

local (sakila) x local (sakila) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

infilink_database

justin

Tables

tagdata

Views

Stored Procedures

Functions

sakila

Tables

actor

address

category

city

country

customer

film

film_actor

film_category

film_text

inventory

Administration Schemas

Information

Table: tagdata

Columns:

tagname varchar(70)

tagval double

strtagval varchar(128)

tagtime datetime

groupname varchar(70)

quality int(11)

Query 1

1 • SELECT * FROM justin.tagdata;

Limit to 1000 rows

Result Grid

tagname	tagval	strtagval	tagtime	groupname	quality
_Second	55	NA	2019-06-17 11:52:55	System	NA
Channel1_System_BaudRate	9600	NA	2019-06-17 11:52:55	OPCTags	NA
Channel1_ST2_1_System_AutoCrea	0	NA	2019-06-17 11:52:55	OPCTags	NA
Channel1_ST2_1_UnsolPckRcvdO_1	1	NA	2019-06-17 11:52:55	OPCTags	NA
Channel1_ST2_1_UnsolPckRcvdOnT	1	NA	2019-06-17 11:52:55	OPCTags	NA
_Second	56	NA	2019-06-17 11:52:56	System	NA
_Second	57	NA	2019-06-17 11:52:57	System	NA
_Second	58	NA	2019-06-17 11:52:58	System	NA
_Second	59	NA	2019-06-17 11:52:59	System	NA
_Second	0	NA	2019-06-17 11:53:00	System	NA
_Second	1	NA	2019-06-17 11:53:01	System	NA
_Second	2	NA	2019-06-17 11:53:02	System	NA

Output

Action Output

Time	Action	Message	Duration / Fetch
1 11:56:45	SELECT * FROM justin.tagdata LIMIT 0, 1000	229 row(s) returned	0.000 sec / 0.000 sec
2 13:06:57	COMMIT	0 row(s) affected	0.000 sec

Object Info Session

Connection cancelled

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

Type here to search

1:37 PM 6/17/2019

Step 10: AWS RDS

Once logged into Amazon Web Services, go to “Create a Database” on the RDS page.

The screenshot displays the Amazon RDS console interface. On the left, a navigation sidebar lists various RDS features: Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations. The main content area is titled 'Amazon RDS' and includes a list of links for 'Allocated storage (0 TB/100 TB)', 'Click here to increase DB instances limit', 'Reserved instances (0/40)', 'Snapshots (1)', 'Manual (1/100)', 'Automated (0)', 'Recent events (6)', 'Event subscriptions (0/20)', 'Default (1)', 'Custom (0/100)', 'Option groups (1)', 'Default (1)', 'Custom (0/20)', 'Subnet groups (1/50)', 'Supported platforms VPC', and 'Default network vpc-3542244f'. Below this, the 'Create database' section explains that Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud, and provides buttons for 'Restore from S3' and 'Create database'. A note states that DB instances will launch in the US East (N. Virginia) region. The 'Service health' section shows the 'Current status' as 'Amazon Relational Database Service (N. Virginia)' with a green checkmark, and 'Details' as '[RESOLVED] Increased API Latencies and Management Console Error Rates'. On the right, a 'Feature Spotlight' section highlights 'RDS Multi-AZ for Read Replicas', 'Aurora Serverless', 'RDS M5 Instances', and 'RDS Performance Insights'. The bottom of the page includes a footer with 'Feedback', 'English (US)', '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', 'Terms of Use', and a system clock showing '3:45 PM 6/17/2019'.

Amazon RDS

Dashboard

- Databases
- Query Editor
- Performance Insights
- Snapshots
- Automated backups
- Reserved instances
- Subnet groups
- Parameter groups
- Option groups
- Events
- Event subscriptions
- Recommendations

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

[Restore from S3](#) [Create database](#)

Note: your DB instances will launch in the US East (N. Virginia) region

Service health

[View service health dashboard](#)

Current status	Details
Amazon Relational Database Service (N. Virginia)	[RESOLVED] Increased API Latencies and Management Console Error Rates

Feature Spotlight

RDS Multi-AZ for Read Replicas

Enable Multi-AZ configurations for RDS Read Replicas to build a more resilient DR strategy. [Learn more](#)

Aurora Serverless

Run an Aurora database without having to manage instances, and pay on a per-second basis for capacity. [Learn more](#)

RDS M5 Instances

Launch the next generation of EC2 General Purpose compute instances when using Amazon RDS. [Learn more](#)

RDS Performance Insights

Feedback English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

3:45 PM 6/17/2019

Step 11: Create your database

Create a MySQL database based on your specifications.

The screenshot shows the AWS RDS console interface. The browser address bar displays the URL: `https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;s3-import=false`. The console header includes the AWS logo, navigation tabs for 'Services' and 'Resource Groups', and user information for 'Kessler Ellis Products' in 'N. Virginia'. A left-hand sidebar lists the steps of the database creation process: 'Step 1: Select engine', 'Step 2: Choose use case', 'Step 3: Specify DB details', and 'Step 4: Configure advanced settings'. The main content area is titled 'Select engine' and features a section for 'Engine options' with six selectable database engines: Amazon Aurora, MySQL (selected), MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. Each option includes a radio button, the engine name, and its logo. Below the MySQL option, there is a detailed description of MySQL on RDS, highlighting its popularity and scalability, followed by a bulleted list of its features: support for database sizes up to 64 TiB, various instance classes (General Purpose, Memory Optimized, and Burstable Performance), automated backup and recovery, and up to 5 Read Replicas per instance.

Select engine

Step 1
Select engine

Step 2
Choose use case

Step 3
Specify DB details

Step 4
Configure advanced settings

Engine options

- ☐ Amazon Aurora
- ☒ MySQL
- ☐ MariaDB
- ☐ PostgreSQL
- ☐ Oracle
- ☐ Microsoft SQL Server

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 5 Read Replicas per instance, within a single Region or cross-region.

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

3:47 PM 6/17/2019

Step 12: Take the endpoint and use it to connect from MySQL workbench

After creating your AWS RDS database instance, you will receive an endpoint under “connectivity and security”. Use this as the host name when you connect to your database in MySQL workbench. Your username and password should be the ones you set up when you created the RDS instance.

The image displays two overlapping windows. The background window is the AWS Management Console, specifically the 'Connectivity & security' page for an Amazon RDS instance. It shows the 'Endpoint & port' section with the following details:

- Endpoint:** kep.coy9czfiurvm.us-east-1.rds.amazonaws.com
- Port:** 3306

The foreground window is the 'Connect to Database' dialog in MySQL Workbench. It is configured with the following settings:

- Stored Connection:** (Empty dropdown)
- Connection Method:** Standard (TCP/IP)
- Parameters tab:**
 - Hostname:** us-east-1.rds.amazonaws.com
 - Port:** 3306
 - Username:** kessler
 - Password:** (Empty field with 'Store in Vault ...' and 'Clear' buttons)
 - Default Schema:** (Empty field)

At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar with the text 'Type here to search', and several application icons including the Task View button, File Explorer, Mail, Edge, and the Microsoft Store.

Step 13: Use the migration wizard to move the Infilink database from your localhost to your RDS instance

This wizard will walk you through moving your Infilink database with your tag table and alarm table into your Amazon RDS database that you are now connected to on MySQL.

