COLUMBIA UNIVERSITY | MAILMAN SCHOOL of PUBLIC HEALTH
BIOSTATISTICS

In-Class Assignment 14

Adding Triggers, Inputting Data, and Creating Views in MySQL

Read the scenario below, then follow the instructions and answer the questions that follow. Add your SQL solutions in the relevant sections of the **In-Class Assignment 14.sql** SQL script and submit to Canvas by the deadline listed above. Save your file frequently to avoid losing work!

Scenario:

You previously built a database for the randomized controlled trial (RCT) examining the efficacy of a new weight-loss drug called WL001A. You will now add triggers for data validation, populate the tables with data, and create views to store summaries of the data. Relevant information about the study is included below to help guide your design of the database.

Data Validation:

Add appropriate triggers to perform the following data validation. Test each condition by entering invalid data values to ensure that the trigger works correctly.

1. In the **participants** table:
   - Treatment group field can only take values 1 or 2
   - Valid age range is 18 to 55 years
   - Prediabetes field can only take values 0 or 1

2. In the **adverse_event_log** table:
   - Valid event dates are between 6/15/2017 and CURDATE()
     (Dates are referred to in the format: **'yyyy-mm-dd'**)

Data Entry:

In the **weight** schema, use SQL statements to enter the following data into each table.

NOTE: you must enter data into certain tables before entering data into others in order to prevent errors related to orphaned records. Give thought to which tables should have data entered first.

HINT: to save some typing, you may copy and paste the values below into your SQL script and update syntax around the values. Remember that **text** and **date** values should be enclosed in single quotes. You may also utilize the find and replace functions in MySQL Workbench as needed.

participants

tx_group	age	height	prediabetes
1	46	1.67	0
2	54	1.83	0
2	49	1.74	1
1	47	1.88	0
1	39	1.57	1

ID does not need to be manually added if it has been set to an auto_increment field

In-Class Assignment 14

visits

participant_id	visit_type	weight
1	0	68.04
2	0	91.63
3	0	81.19
1	4	65.77
4	0	102.05
2	4	92.08
5	0	73.49
1	8	64.86

If you get errors when entering the participant IDs, check to see the automatically generated ID values in the participants table and adjust your data entry in this table to match those values

Again, the visit ID will be generated automatically as you enter these records

adverse_event_log

visit_id	adverse_event_id	adverse_event_date
4	3	2017-08-29
5	1	2017-09-30
8	3	2017-11-12

adverse_events

adverse_event_id	adverse_event_type
1	Excessive fatigue
2	Reaction to medication
3	Abnormal lab results
4	Hospitalization
5	Skin rash
6	Other

Although adverse_event_id may be an auto_increment field, you can also manually add values to that field

current_meds

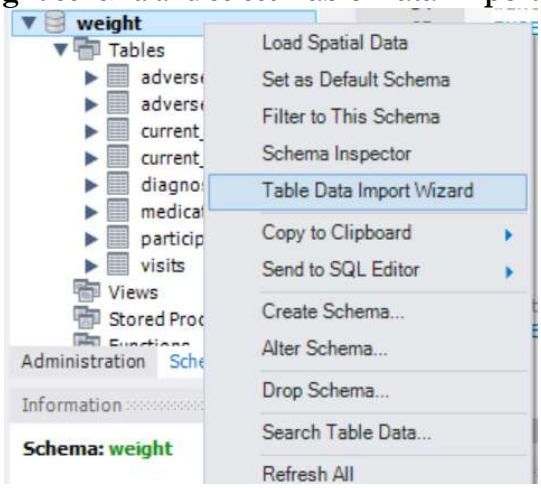
visit_id	medication_id
1	6
1	7086
2	371
3	4241
4	6
5	3500
6	371
7	4241
8	6
8	7086

In-Class Assignment 14

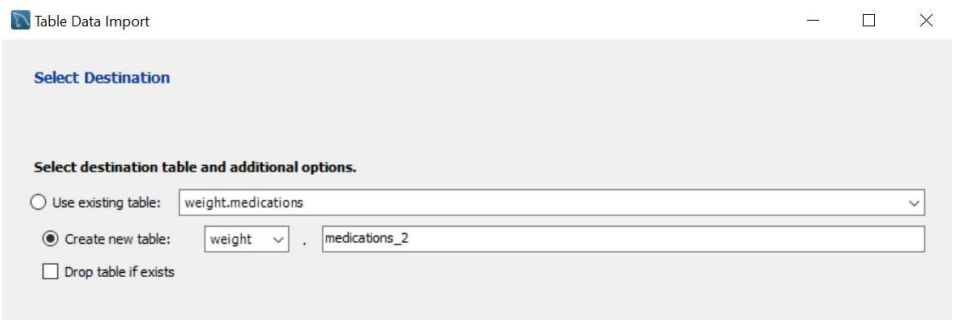
medications

To import the medications csv file, follow the steps below:

1. Right-click the **weight** schema and select **Table Data Import Wizard**



2. Follow prompts to browse to medications.csv file, then select the following option to add the table to the schema as a NEW table called medications_2



3. Accept all remaining defaults and import the table. This may take a couple of minutes due to the size of the csv file.

4. Copy the data from medications_2 into the medications table (this is done so that you don't have to re-define the relationships that already exist with the medications table). To copy data from one table to another, use the following syntax:

```
INSERT INTO table_name (column1, column2, …)
        SELECT column1, column2, …
        FROM old_table;
```

5. Drop the medications_2 table from the schema

In-Class Assignment 14

current_dxs

visit_id	diagnosis_code
1	470
1	460
1	462
3	515
3	471
4	470
4	460
7	477
8	470
8	460

diagnoses

diagnosis_code	diagnosis_name
460	Acute nasopharyngitis (common cold)
461	Acute sinusitis
462	Acute pharyngitis
463	Acute tonsillitis
464	Acute laryngitis and tracheitis
465	Acute upper respiratory infections of multiple or unspecified sites
466	Acute bronchitis and bronchiolitis
470	Deviated nasal septum
471	Nasal polyps
472	Chronic pharyngitis and nasopharyngitis
473	Chronic sinusitis
474	Chronic disease of tonsils and adenoids
475	Peritonsillar abscess
476	Chronic laryngitis and laryngotracheitis
477	Allergic rhinitis
478	Other diseases of upper respiratory tract
510	Empyema
511	Pleurisy
512	Pneumothorax
513	Abscess of lung and mediastinum
514	Pulmonary congestion and hypostasis
515	Postinflammatory pulmonary fibrosis

In-Class Assignment 14

Views:

Create the following views to summarize the data you have entered:

1. **weight_by_visit:** show the weight for each patient at each visit. Show the visit type label instead of its numeric code, as below. Sort by patient ID and visit order.
   0 = Baseline
   4 = Week 4
   8 = Week 8
   12 = Week 12
   16 = Week 16
   20 = Week 20

2. **meds_per_visit:** give the participant ID and number of unique medications they are taking at each visit. As above, show the visit type label instead of its numeric code. Display the results showing highest to lowest count within each patient. Be sure to include counts of zero.

3. **diagnosis_summary:** list all of the diagnoses (by name) in the lookup table and give a count of how many times the diagnosis was recorded for a patient (not necessary to account for unique patients or unique visits). Display from most to least common, and include counts of zero.