Project: New Haven Urgent Care **Team#4**

Test Date: 12-12-2019

Test Case ID#: 3 **Name(s) of Tester(s): Jian Wang**

Test Description (What are you testing? – you must be specific):
the system document when a person does not have insurance

NOTE: The following information must be provided to be given credit for any test.

Test Data (Provide the file name of the script used to insert data, provide a screen capture to reflect data, or provide script here):
No Insurance:

```
insert into PATIENT
values ('tody', null, 'wmag', 'Minnesota', null, null, '1212234', '2010-6-11', '10');
```

Having Insurance:

```
insert into PATIENT
values ('1','June', null, 'Amdrew', 'Minnesota', null, null, '1212235', '2010-7-11', '10');
```

```
insert into PARENT_GUARDIAN
values ('11','papa','p','pe','washton', null,null);
```

```
insert into RELATE
```

values ('11','1');

Insert into INSURANCE

values ('111', '11', '2019-12-12', '1');

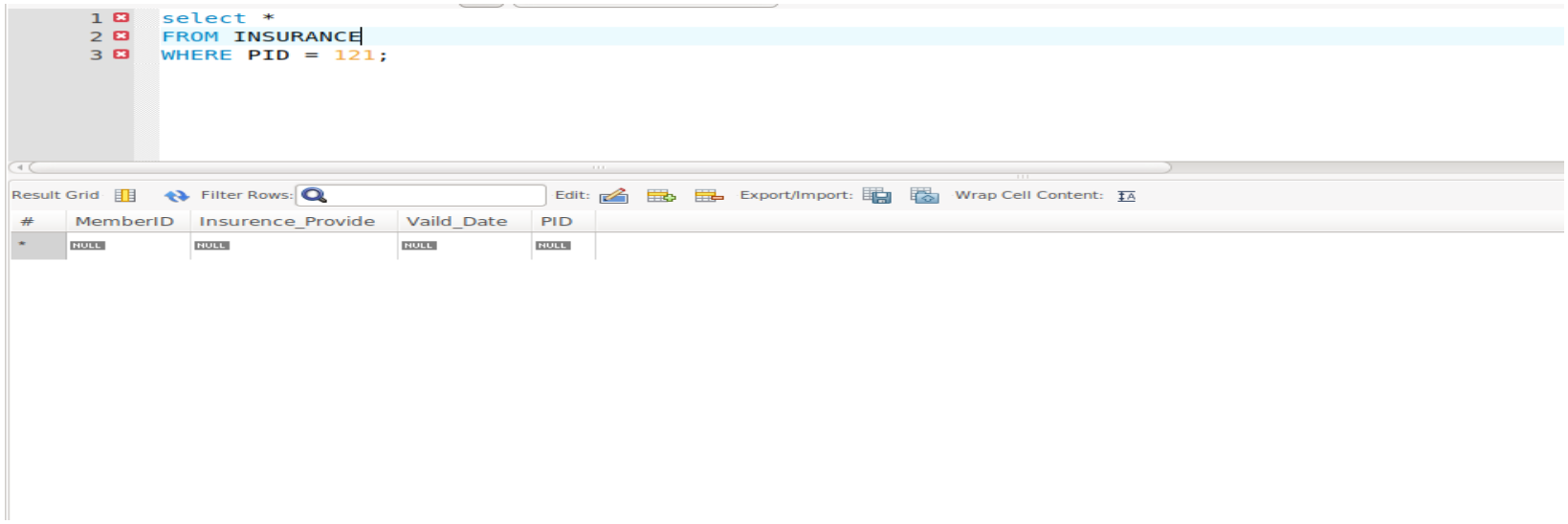
SQL Query(s) used for testing:

select *

FROM INSURANCE

where PID = 121;

The graph below shows the patient whose patient id is 121, he don’t have a insurance. The system return null.

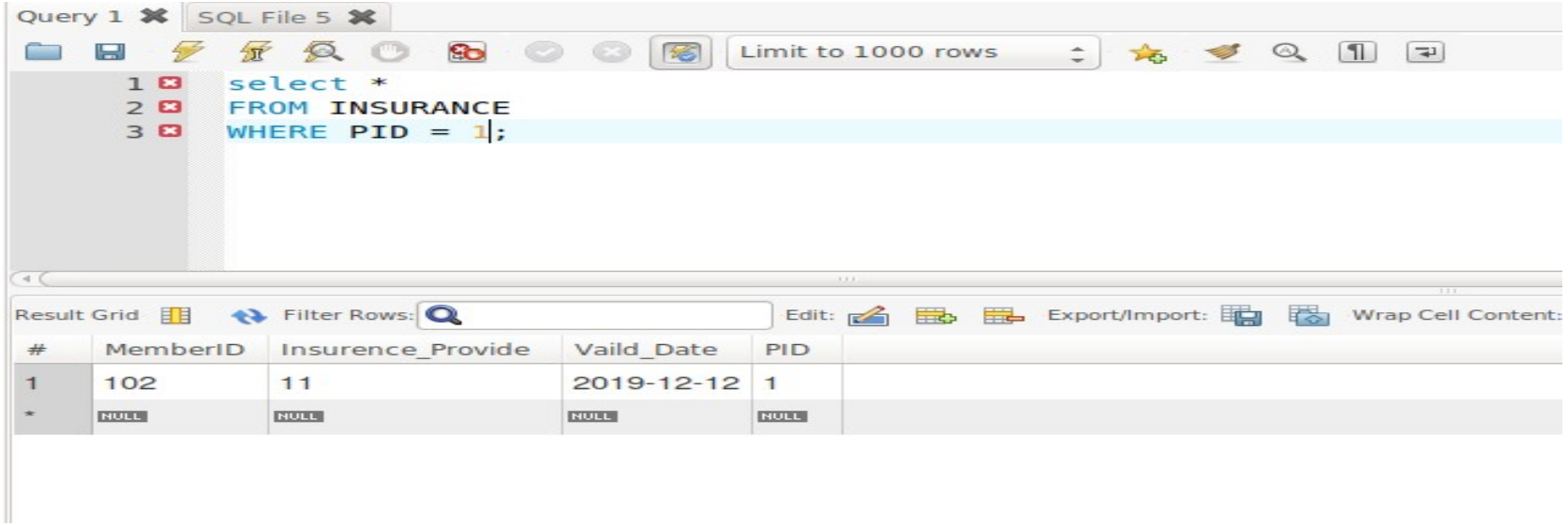


select *

FROM PATIENT

WHERE PID = 1;

this graph shows that the patient whose patient id is 1, he have insurance. The system return INSURANCE table.



Our design has satisfy the requirement. There are two situations in this case. One is no insurance, another one is have insurance. For no insurance, I simply insert a patient who name is tody to PATIENT table without any other information such as parent/guardian, insurance and relate. Therefore, there is only null will be returned.

For having insurance case, there several foreign keys need to complete. Table INSURANCE need a foreign key from Table PARENT/GUARDIAN. And table RELATE also have foreign key form PARENT and PATIENT. So, I insert PATIENT and PARENT/GUARDIAN's data first, due to they have no foreign key. Then I insert RELATE and INSURANCE. This time, all foreign keys are complete so it return a table with data.