1- Assumptions

Assumptions:

Assumption 1: Patient Health Records Association

Assumption: Each patient can have multiple health records, and there is a one-to-many relationship between patients and health records.

Reasoning: The design includes a HealthRecord table with a foreign key referencing the Patient table.

Assumption 2: Medications Association

Assumption: Nurses can administer multiple medications to multiple patients, and there is a many-to-many relationship between nurses and patients through the Medication table.

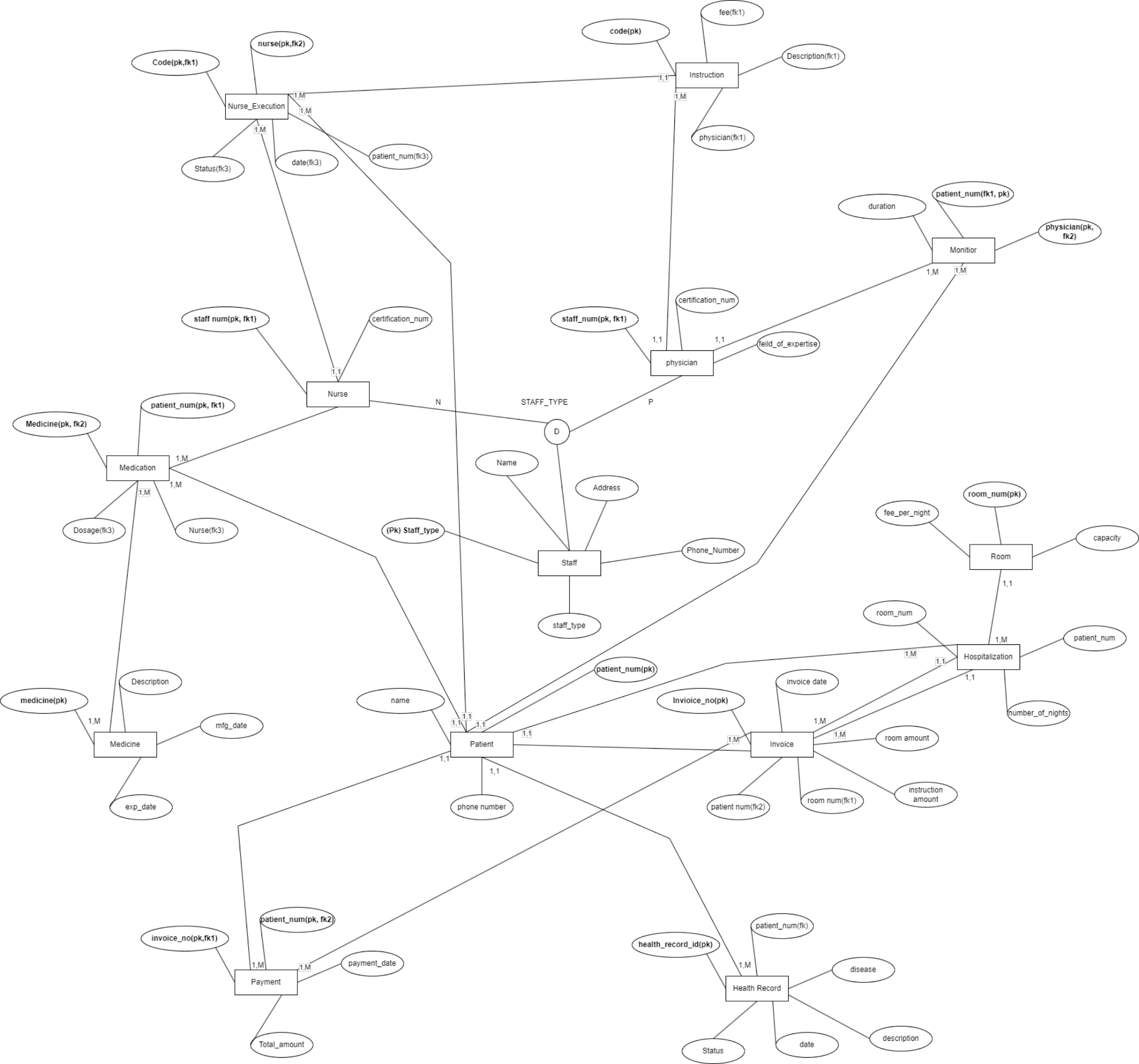
Reasoning: The Medication table has foreign keys referencing both the Nurse and Patient tables.

Assumption 3: Room Occupancy Logic

Assumption: Room occupancy is calculated based on the presence of patients in the room.

Reasoning: The RoomOccupancyView is created to provide a dynamic view of room occupancy based on patient records.

2- (E)ERD



3- Relations and keys

Relation Physician:

Attributes: Physician\_id, Name, Field\_of\_expertise, address, Certificate\_No, Phone\_number

Primary Key: {Physician\_id}

Relation Rooms:

Attributes: Room\_id, Fee\_per\_night, Capacity

Primary Key: {Room\_id}

Relation Patient:

Attributes: Patient\_id, Phone\_number, Address, Name, Rooms\_Room\_id

Primary Key: {Patient\_id}

Foreign Key: {Rooms\_Room\_id references Rooms(Room\_id)}

Relation HealthRecord:

Attributes: Patient\_id, status, Date, description, Disease, Patient\_Patient\_id

Primary Key: {Patient\_id}

Foreign Key: {Patient\_Patient\_id references Patient(Patient\_id)}

Relation Nurse:

Attributes: Nurse\_id, Name, Certificate\_number, phone\_number, address, Physician\_Physician\_id

Primary Key: {Nurse\_id}

Foreign Key: {Physician\_Physician\_id references Physician(Physician\_id)}

Relation Payment:

Attributes: Patient\_id, date, amount, Patient\_Patient\_id, Rooms\_Room\_id

Primary Key: {Patient\_id}

Foreign Keys: {Patient\_Patient\_id references Patient(Patient\_id), Rooms\_Room\_id references Rooms(Room\_id)}

Relation Instruction:

Attributes: Instruction\_id, execution\_status, Fee, Description, Payment\_Patient\_id

Primary Key: {Instruction\_id}

Foreign Key: {Payment\_Patient\_id references Payment(Patient\_id)}

Relation Medication (assuming this should be corrected to Medication):

Attributes: Nurse\_Nurse\_id, Patient\_Patient\_id, Name, Quantity

Primary Key: {Nurse\_Nurse\_id, Patient\_Patient\_id}

Foreign Keys: {Nurse\_Nurse\_id references Nurse(Nurse\_id), Patient\_Patient\_id references Patient(Patient\_id)}

Relation Physician\_has\_Patient:

Attributes: Physician\_Physician\_id, Patient\_Patient\_id

Primary Key: {Physician\_Physician\_id, Patient\_Patient\_id}

Foreign Keys: {Physician\_Physician\_id references Physician(Physician\_id), Patient\_Patient\_id references Patient(Patient\_id)}

4. Views and description

Views:

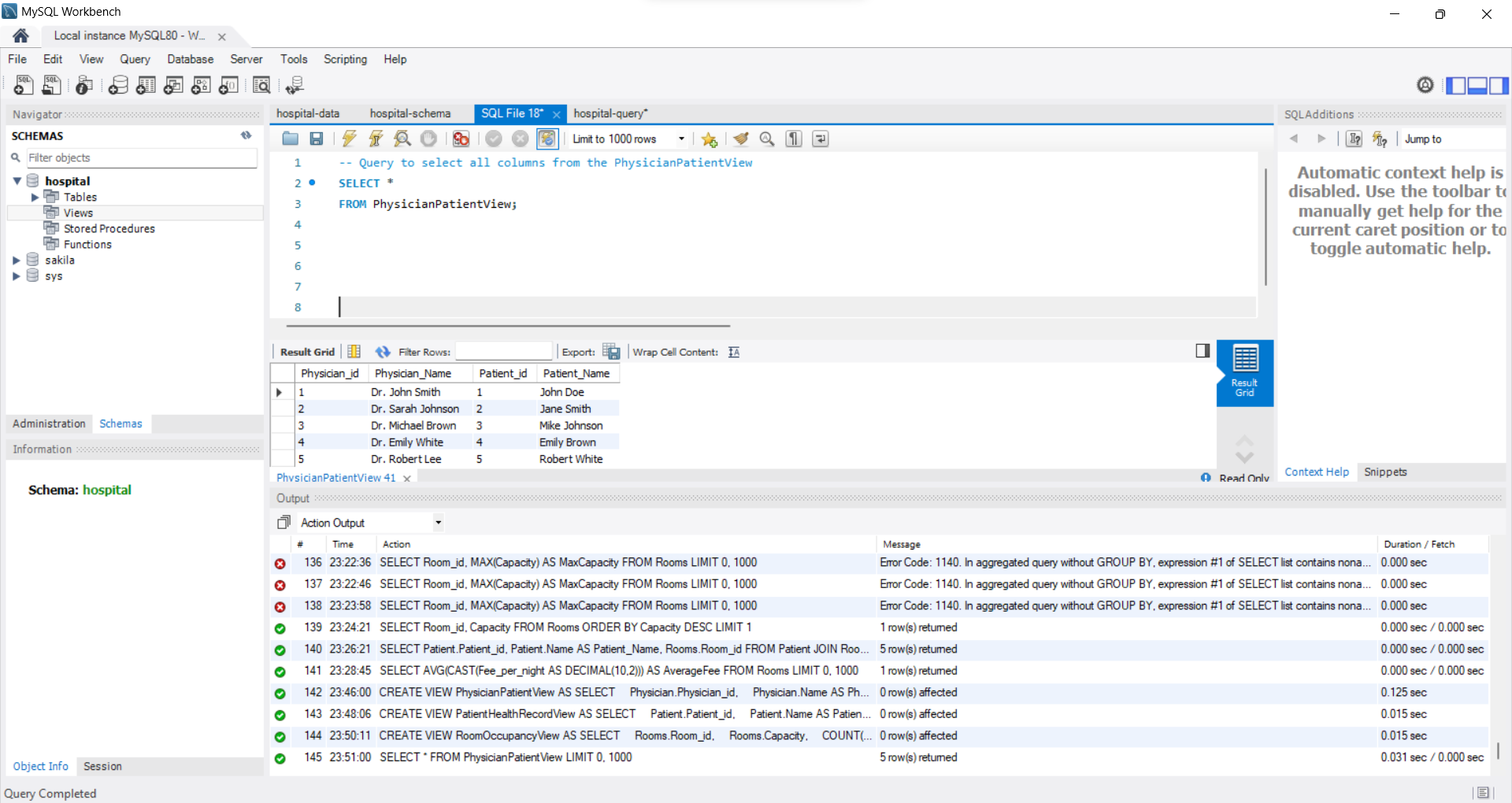
PhysicianPatientView:

Description: This view displays the association between physicians and their assigned patients, including patient names.

Body: A join of Physician, Physician\_has\_Patient, and Patient tables.

Usefulness Discussion: Useful for quick access to the patient-physician association without

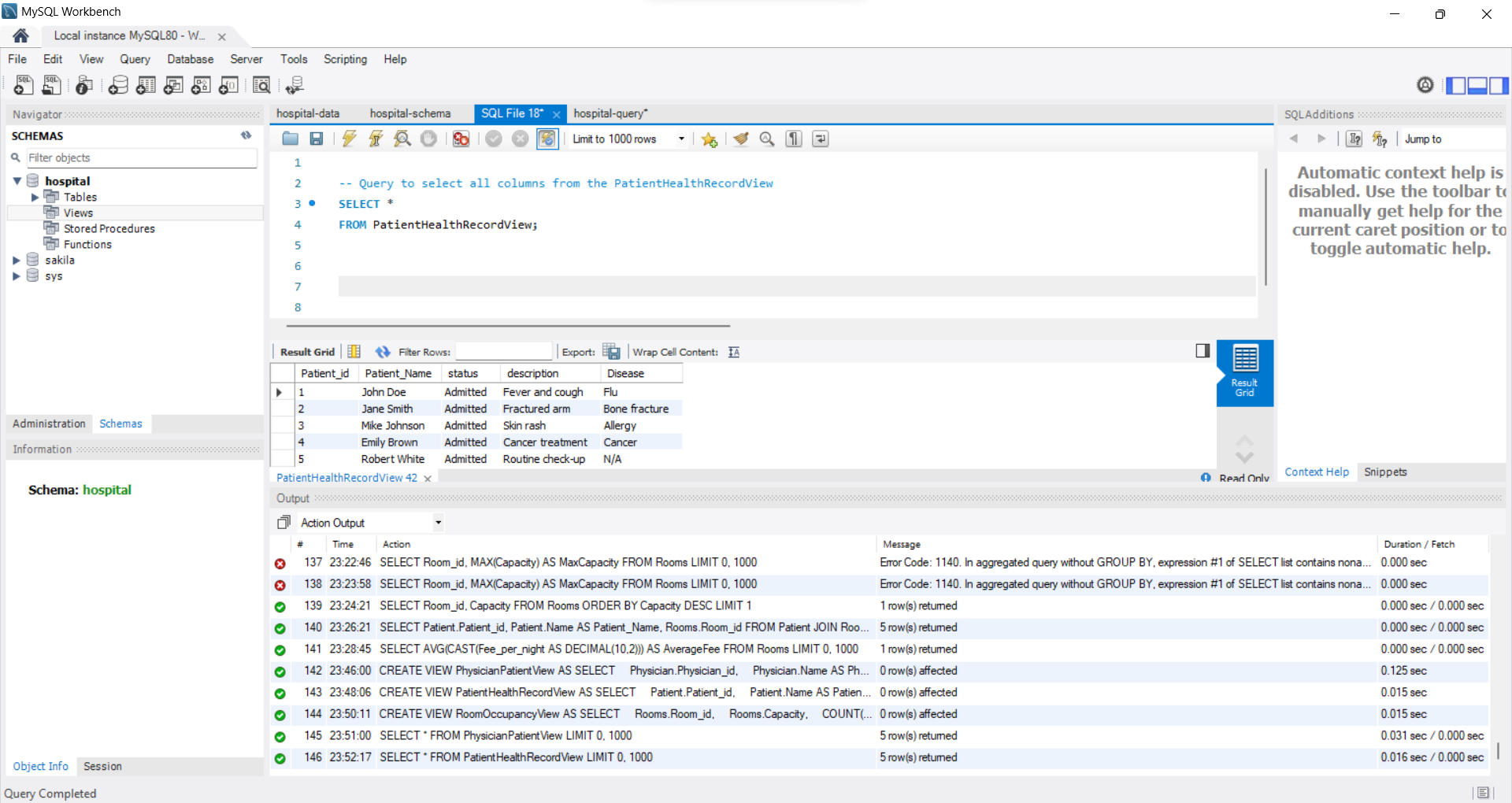
directly querying the join tables.



PatientHealthRecordView:

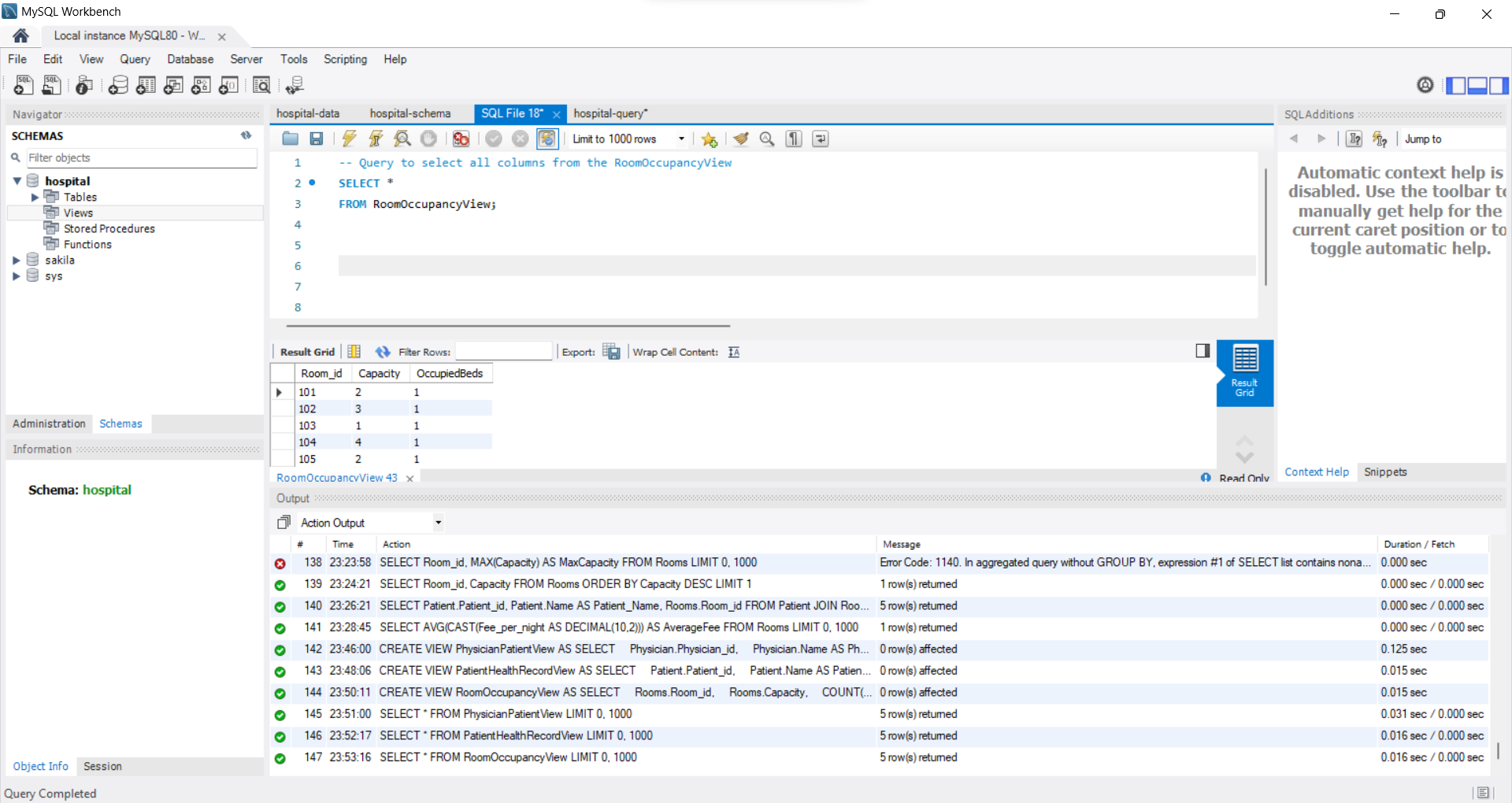
Description: This view shows patients and their associated health records, including status, description, and disease.

Body: A join of Patient and HealthRecord tables.

Usefulness Discussion: Helpful for obtaining a comprehensive view of patient health records in one query.

RoomOccupancyView:

Description: Displays room occupancy, showing the room ID, its capacity, and the number of currently occupied beds.Body: A join of Rooms and Patient tables with aggregation to count occupied beds.

Usefulness Discussion: Offers a quick overview of current room occupancy status.

5- Triggers and description

Triggers:

InsertPatientTrigger:

Description: Updates RoomOccupancyView after a new patient is admitted.

Body: After insert trigger updating the RoomOccupancyView.

Usefulness Discussion: Ensures real-time monitoring of room occupancy without the need for separate queries.

DeletePatientTrigger:

Description: Updates RoomOccupancyView after a patient is discharged.

Body: After delete trigger updating the RoomOccupancyView.

Usefulness Discussion: Keeps room occupancy information up-to-date when patients are discharged.

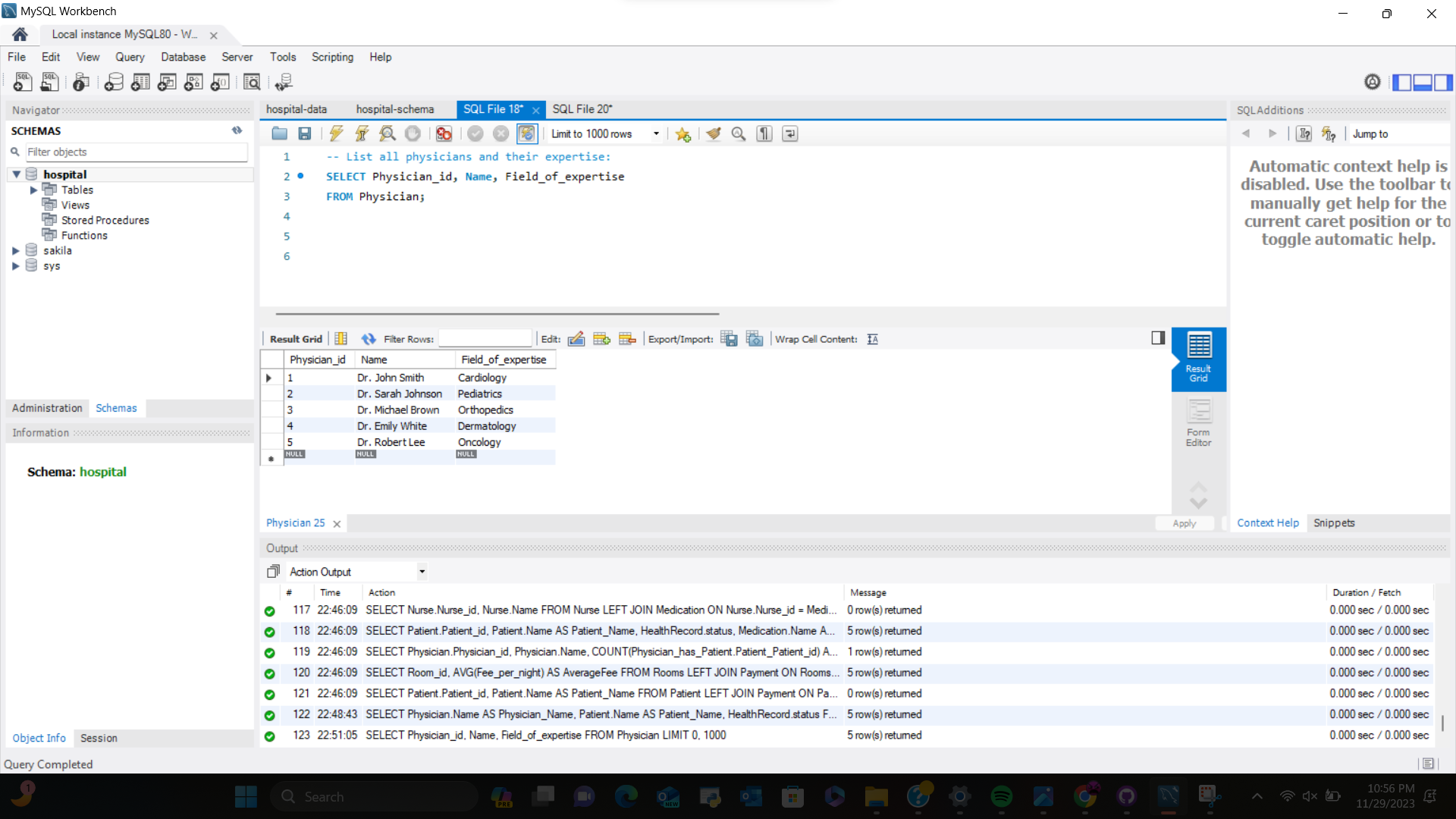
HealthRecordUpdateTrigger:

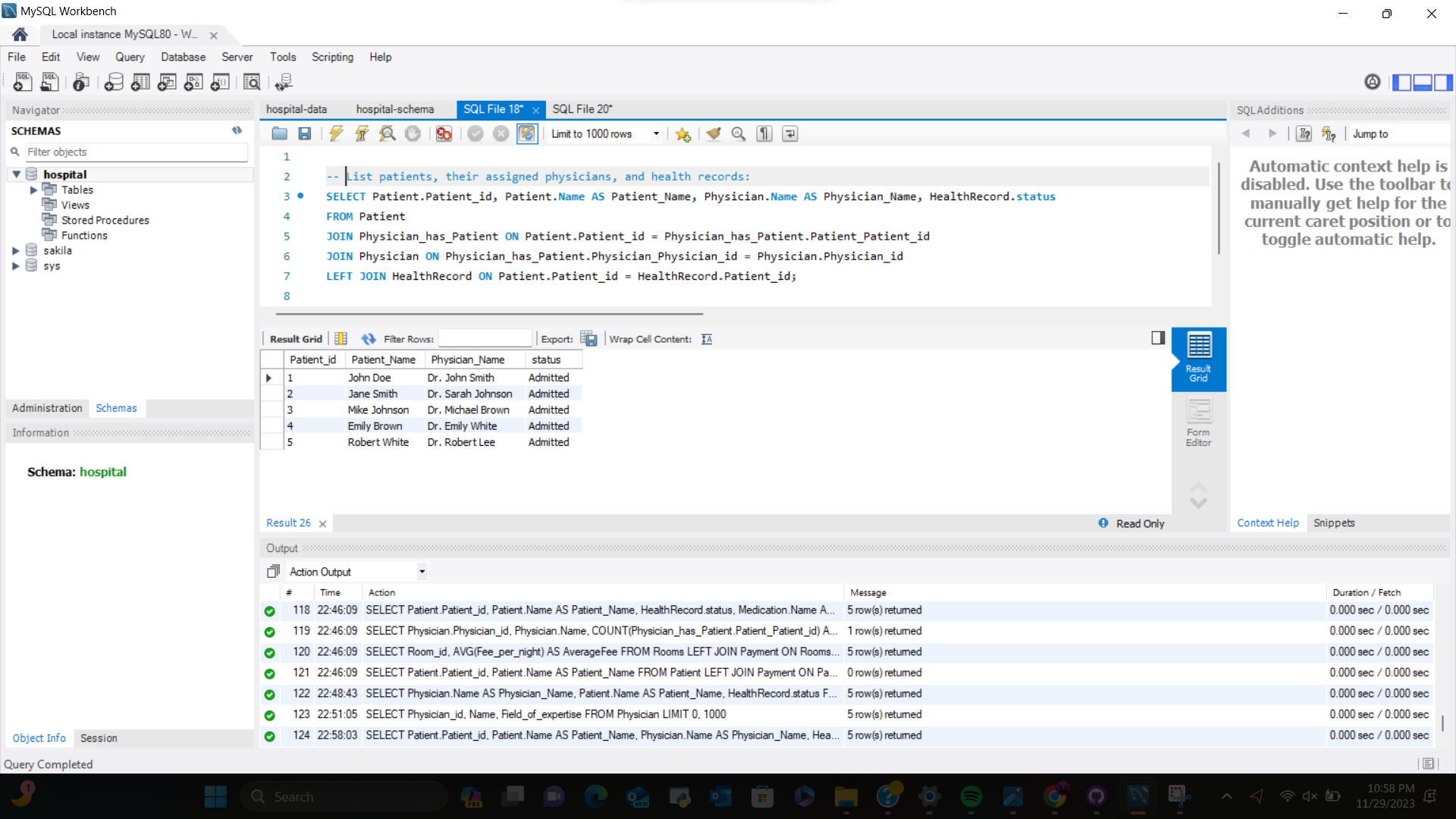
Description: Updates PatientHealthRecordView after a health record is modified.

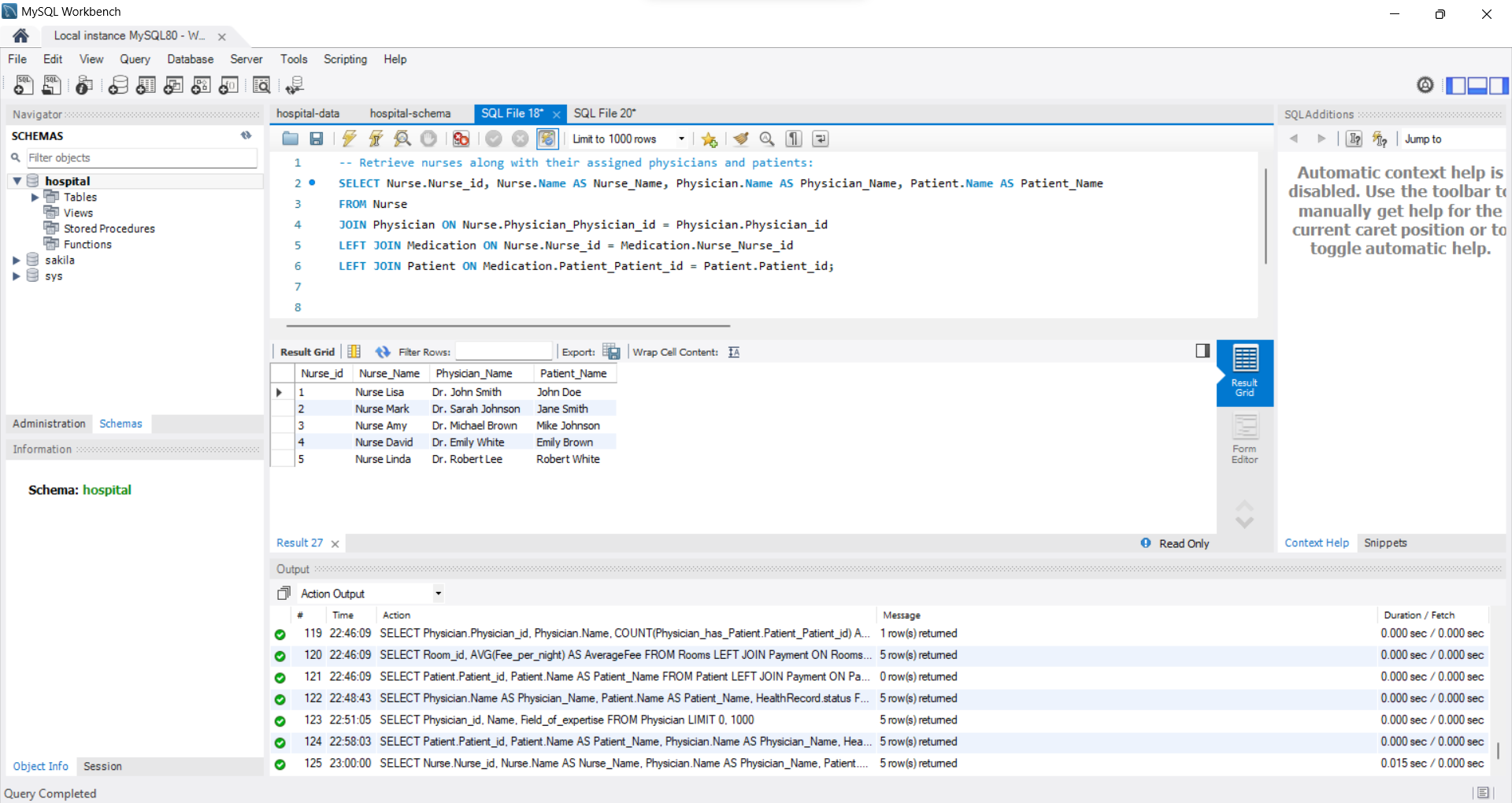
Body: After update trigger updating the PatientHealthRecordView.

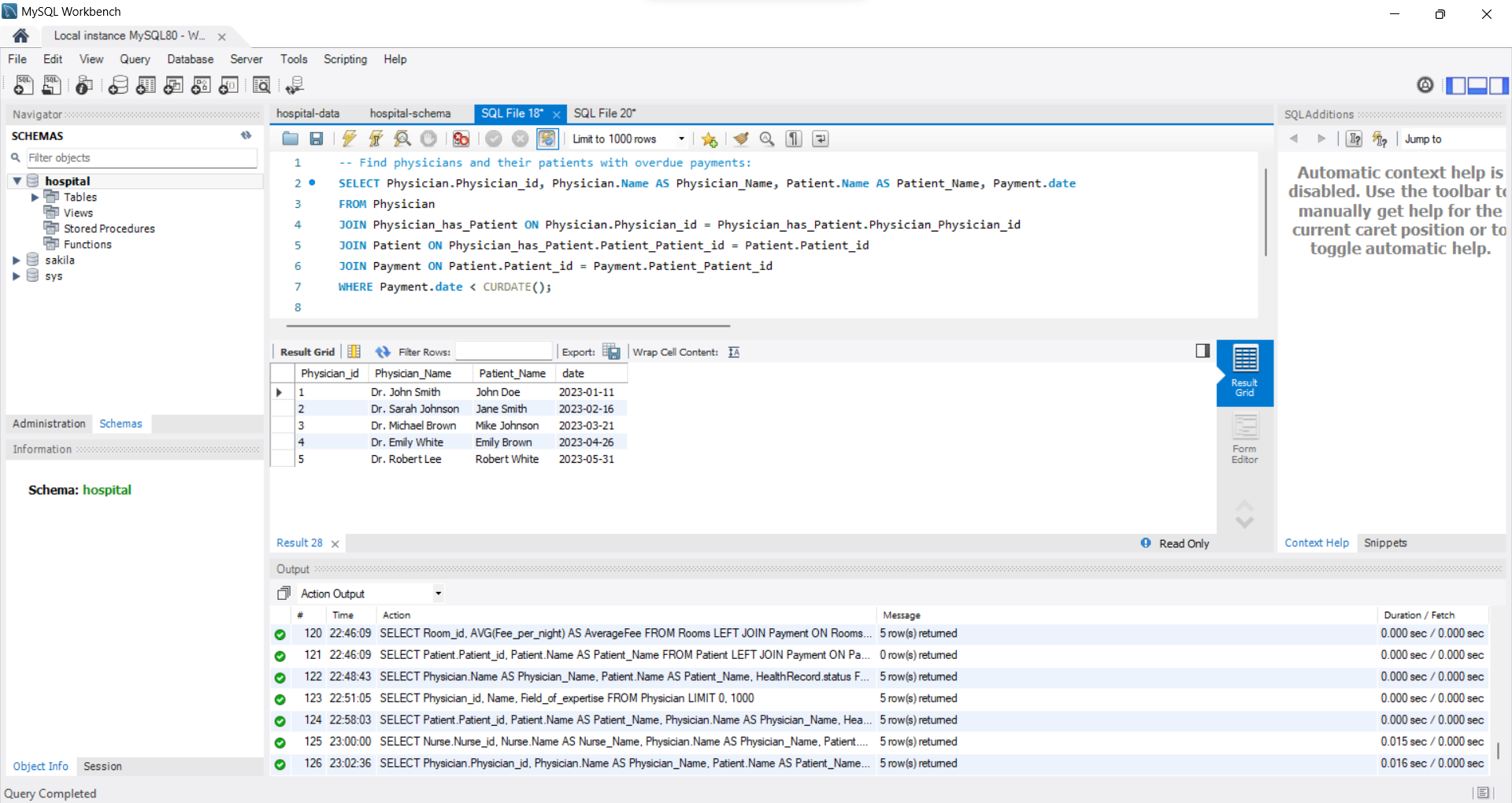
Usefulness Discussion: Maintains consistency in the PatientHealthRecordView after any changes in health records.

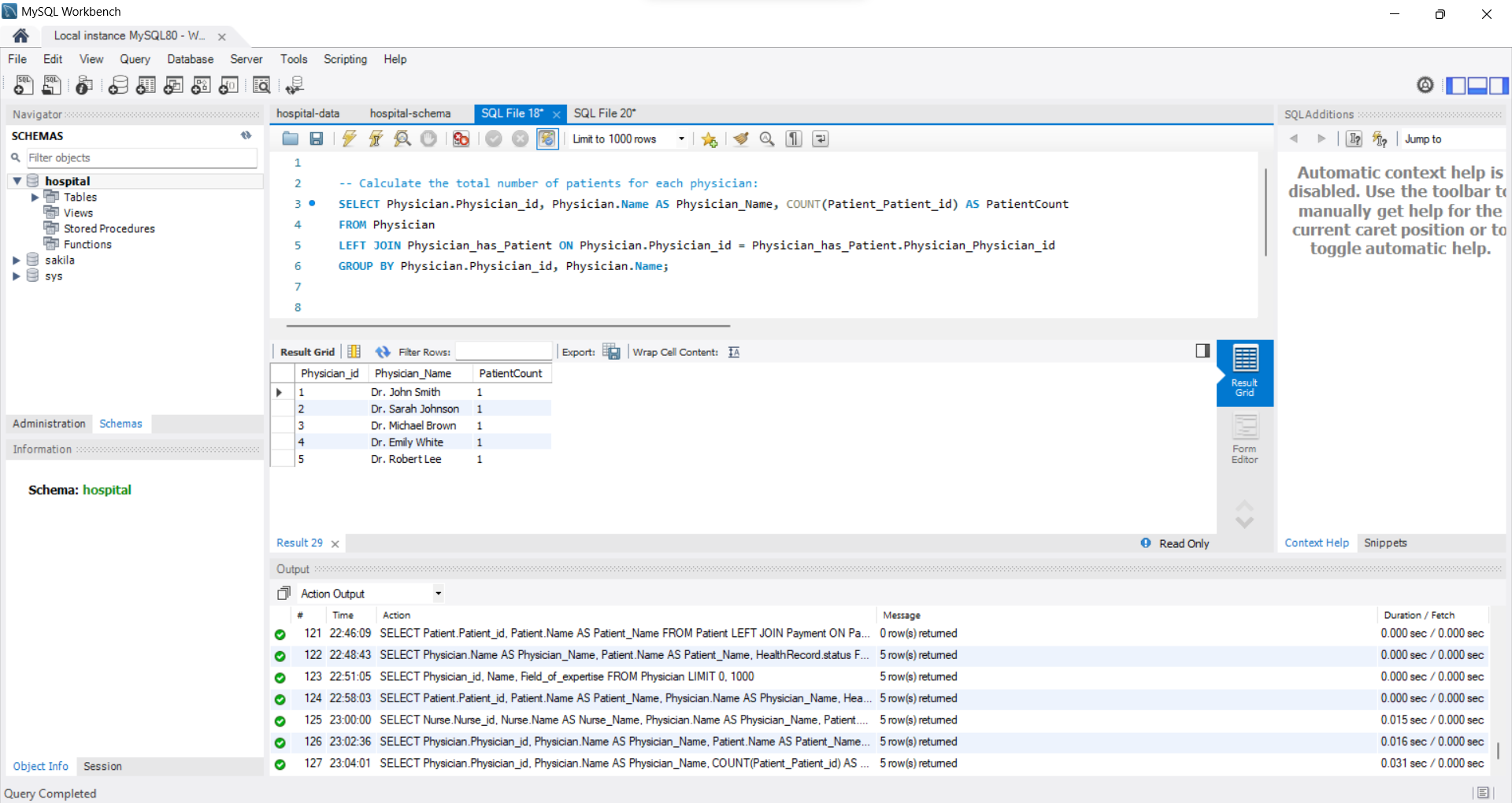
6-Queries, descriptions, and results.

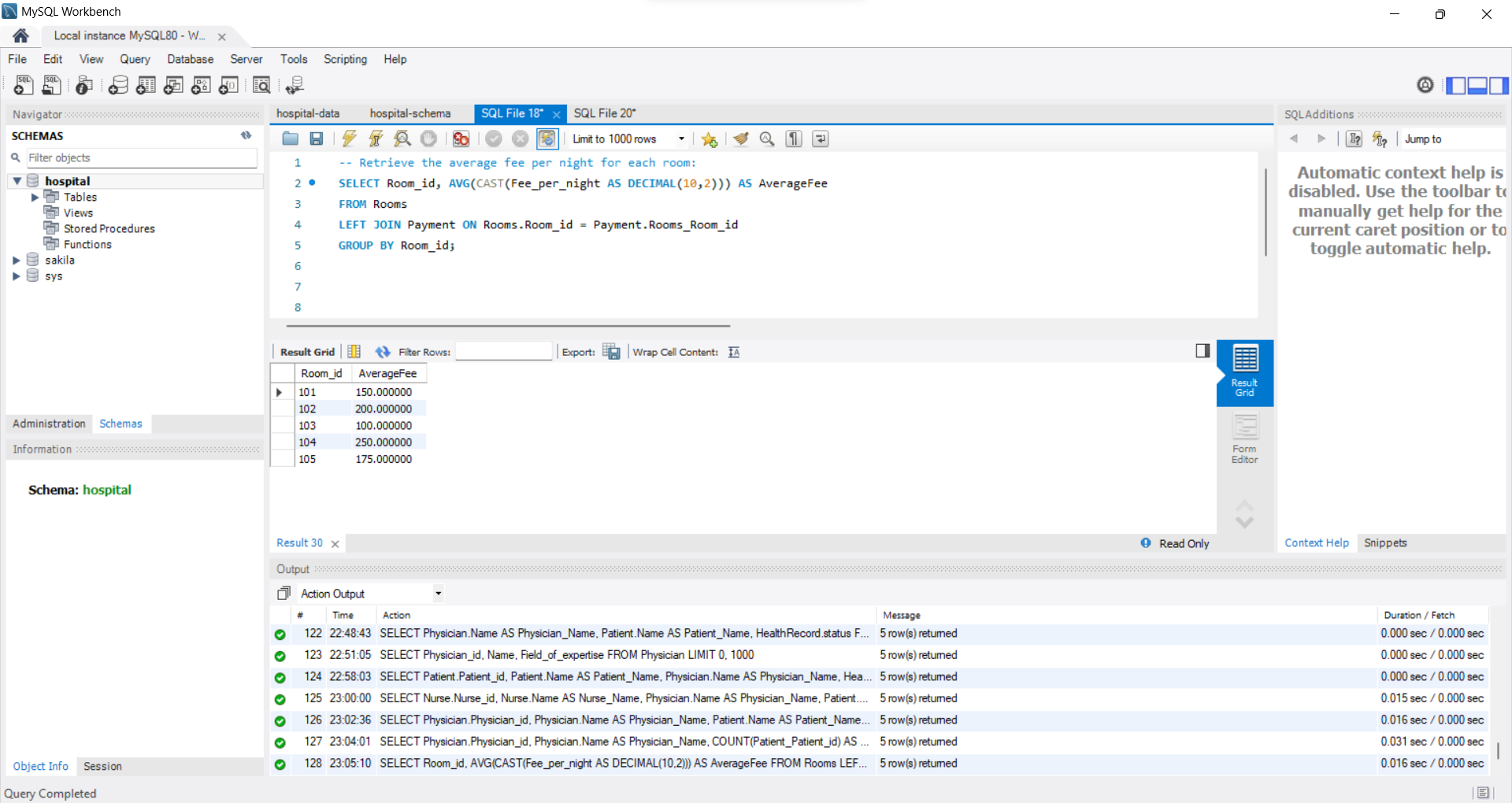


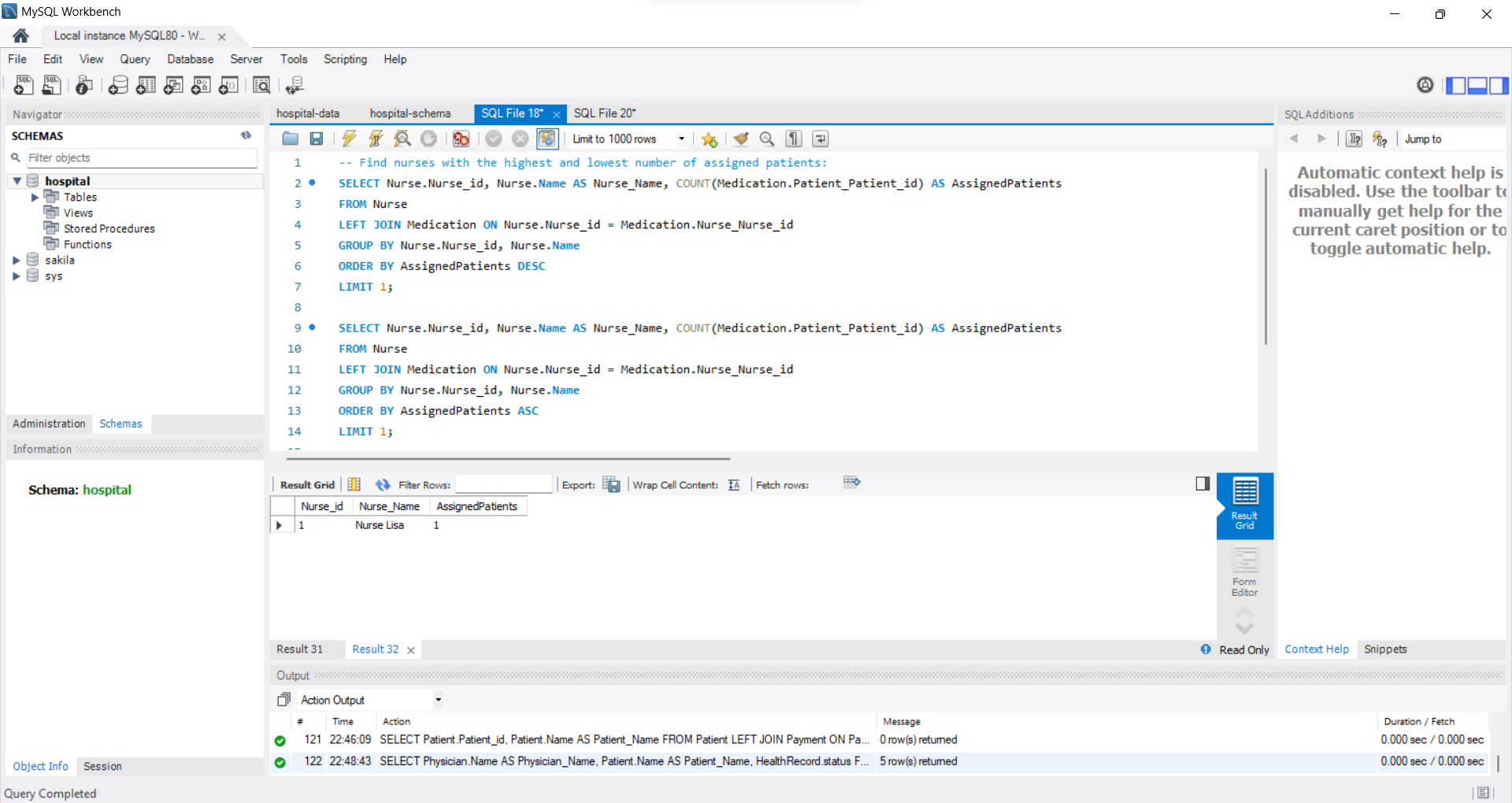


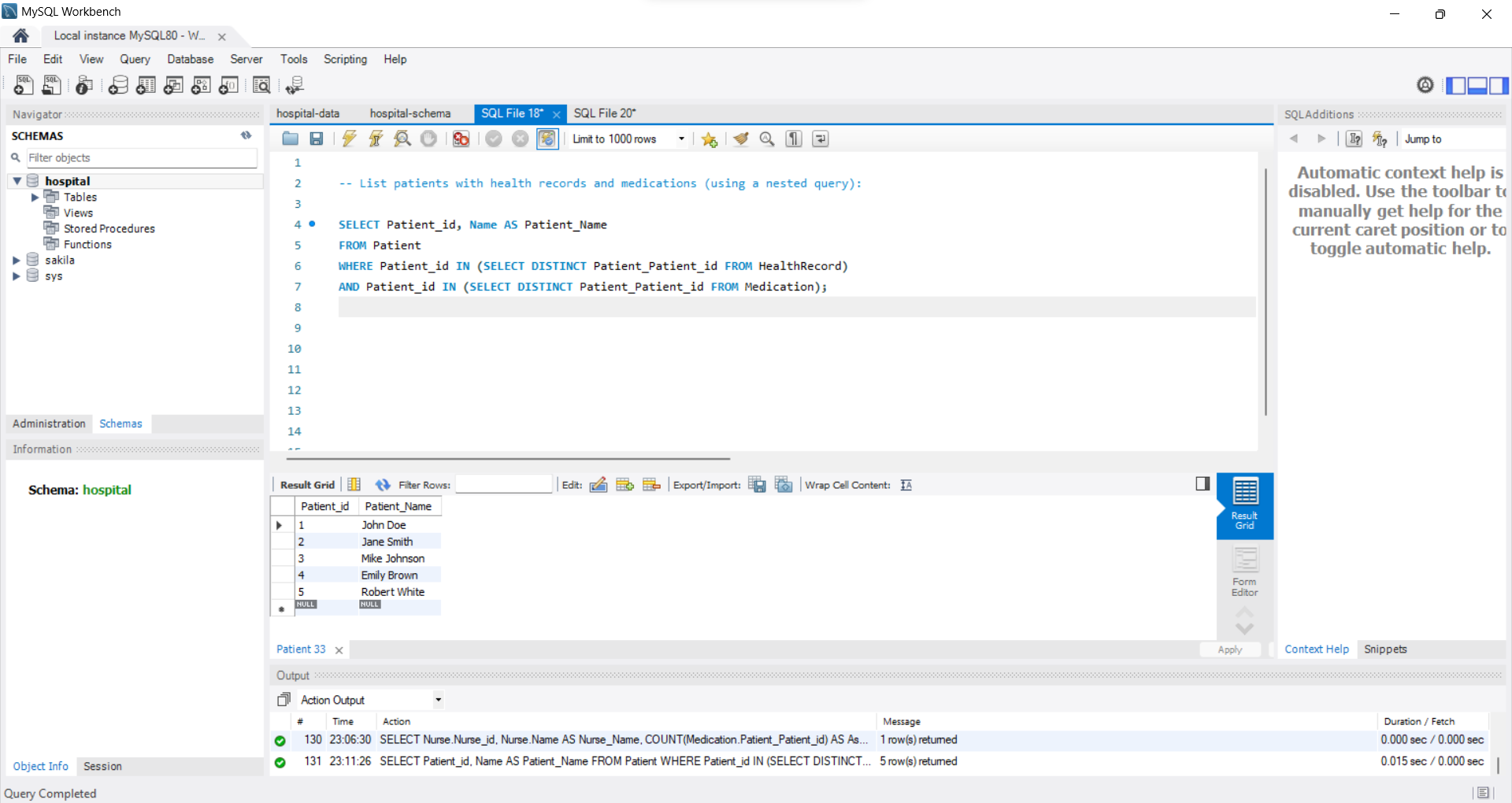


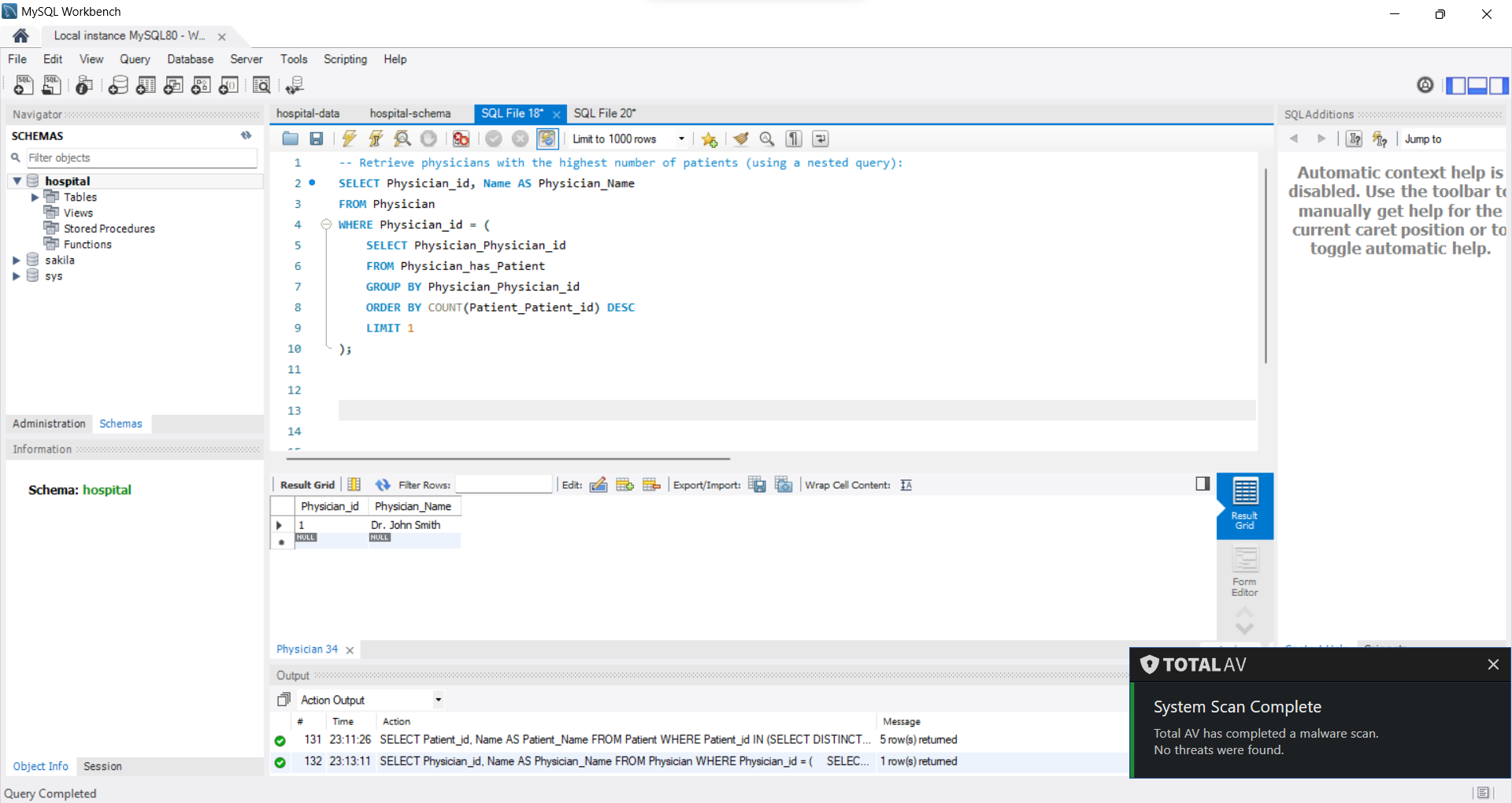


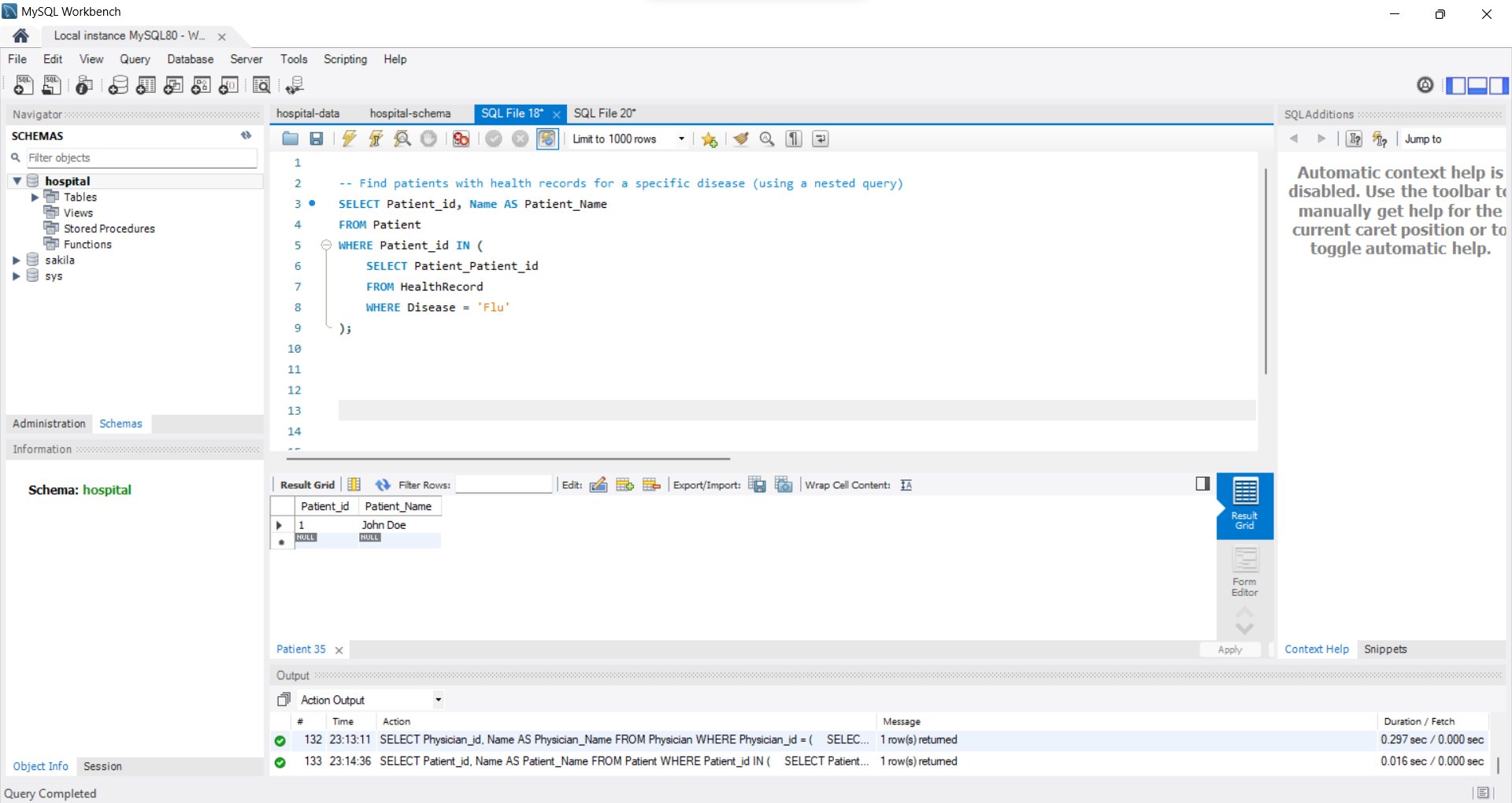


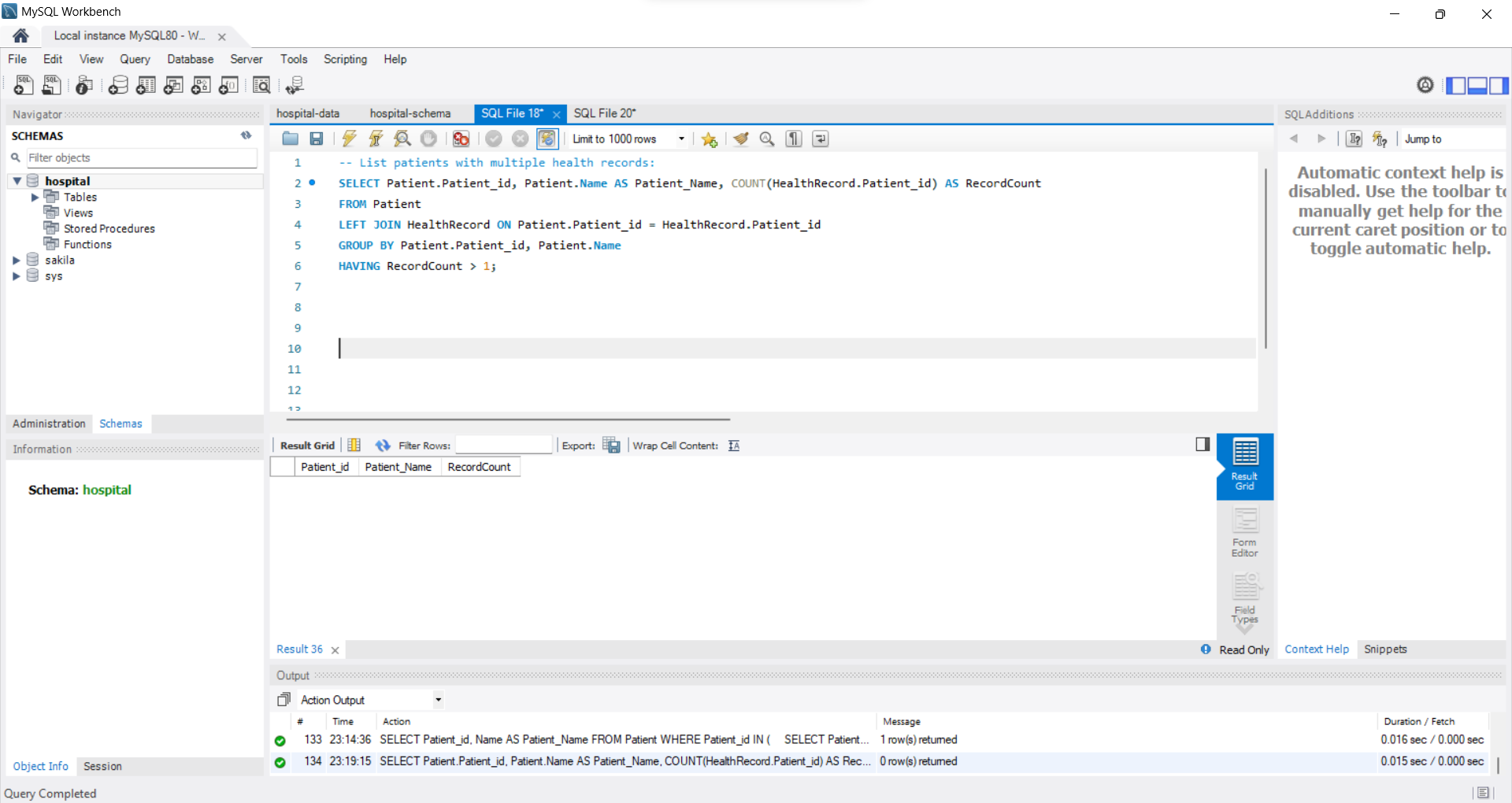


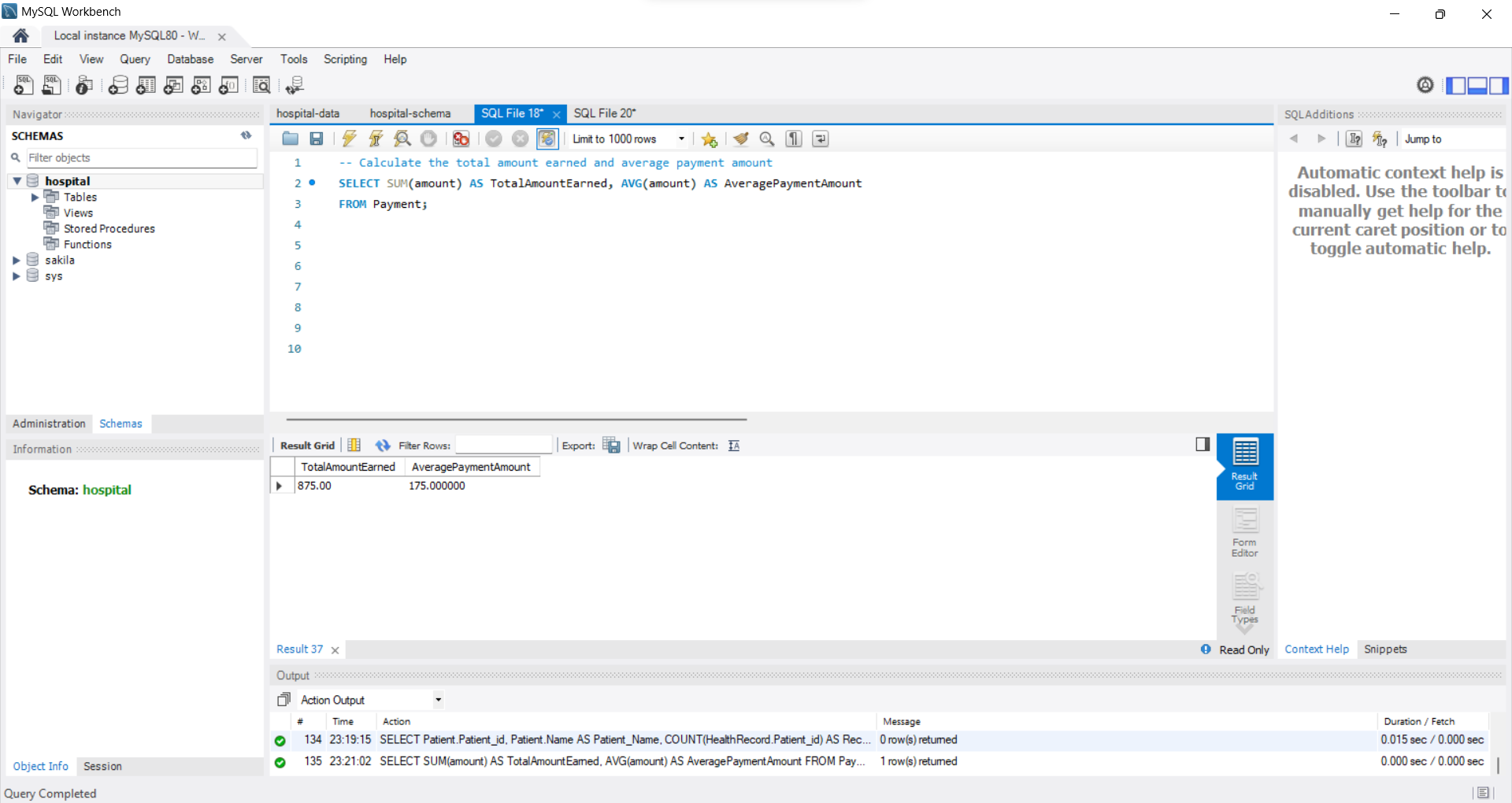


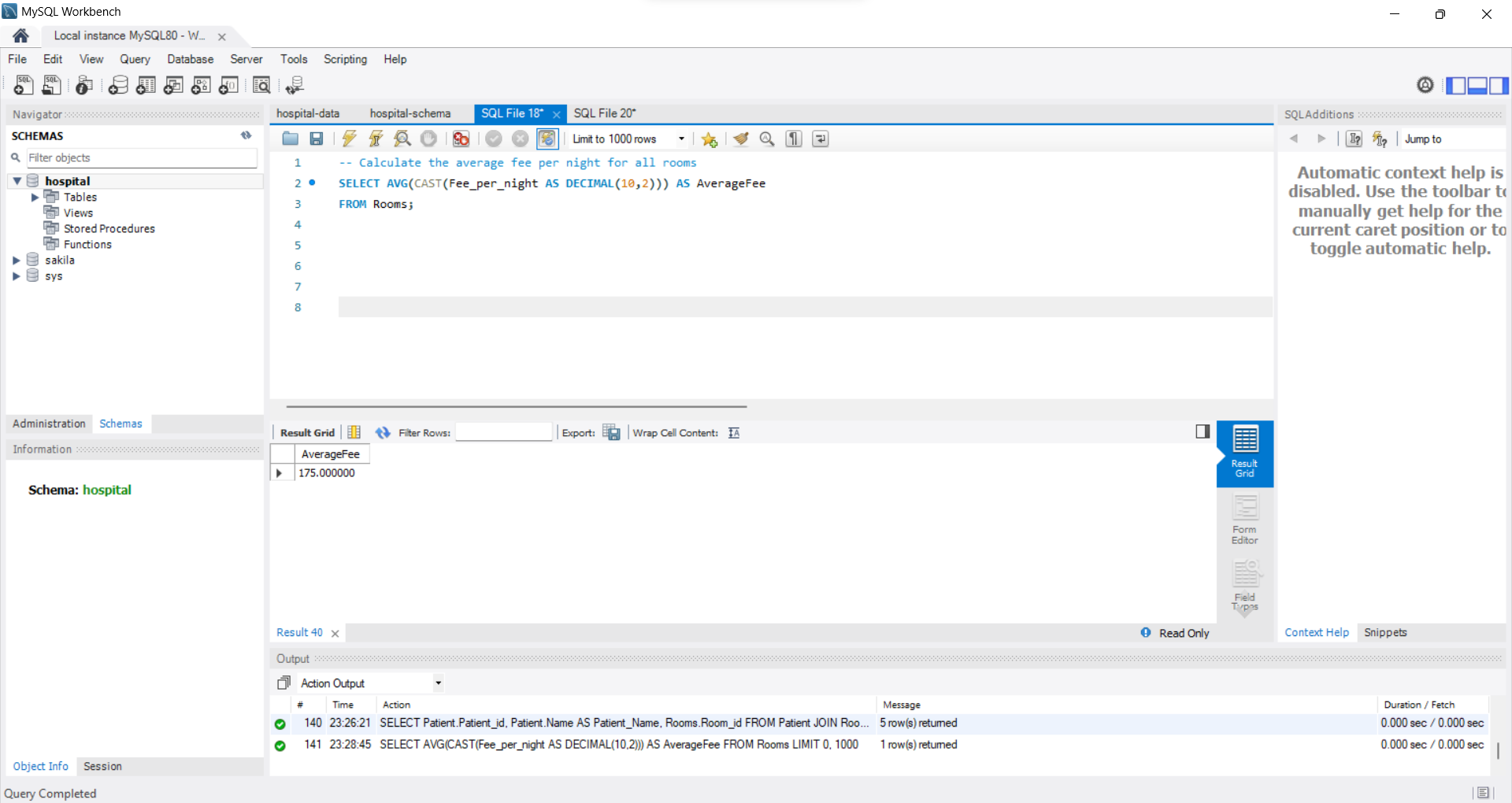


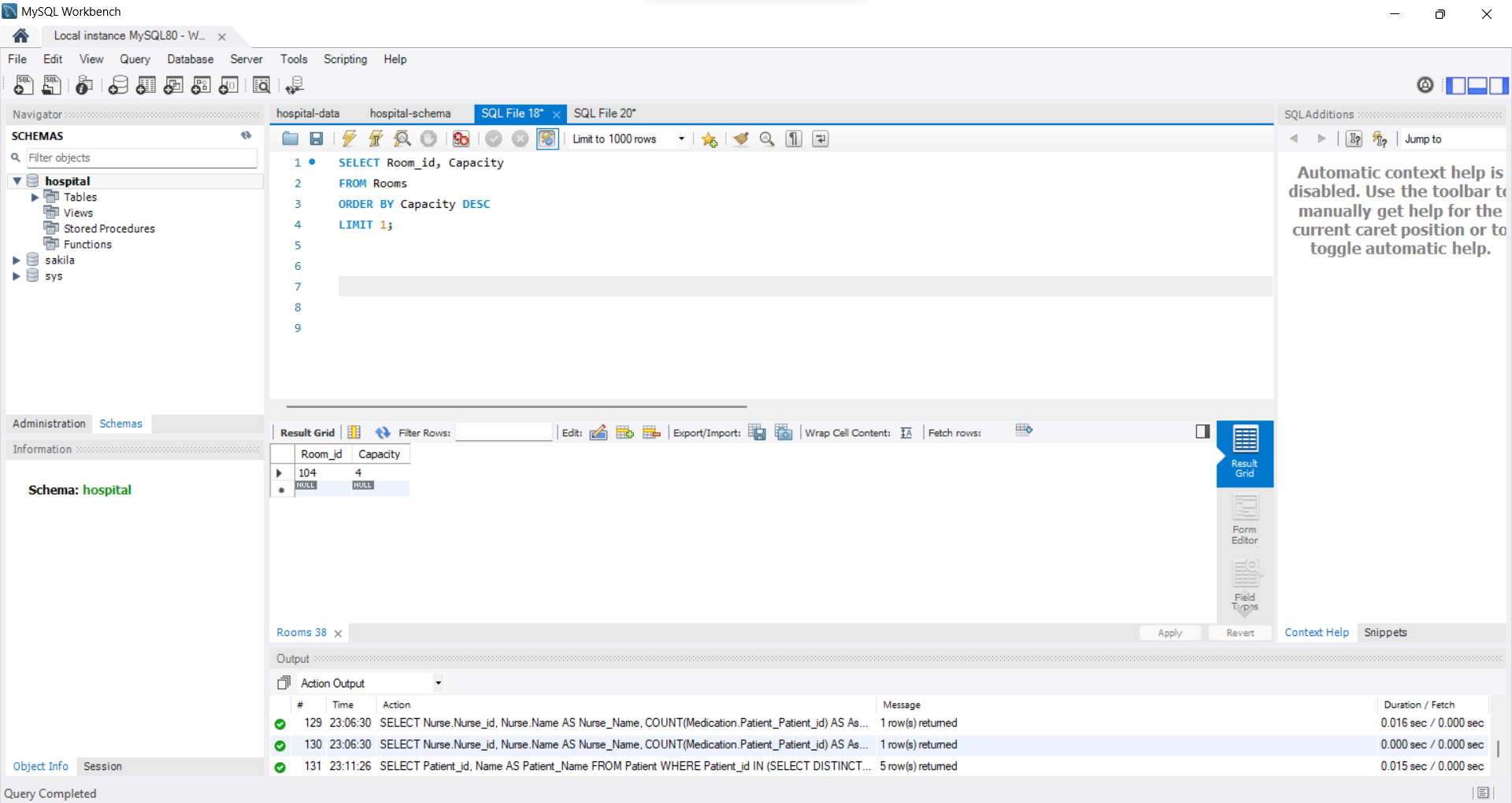


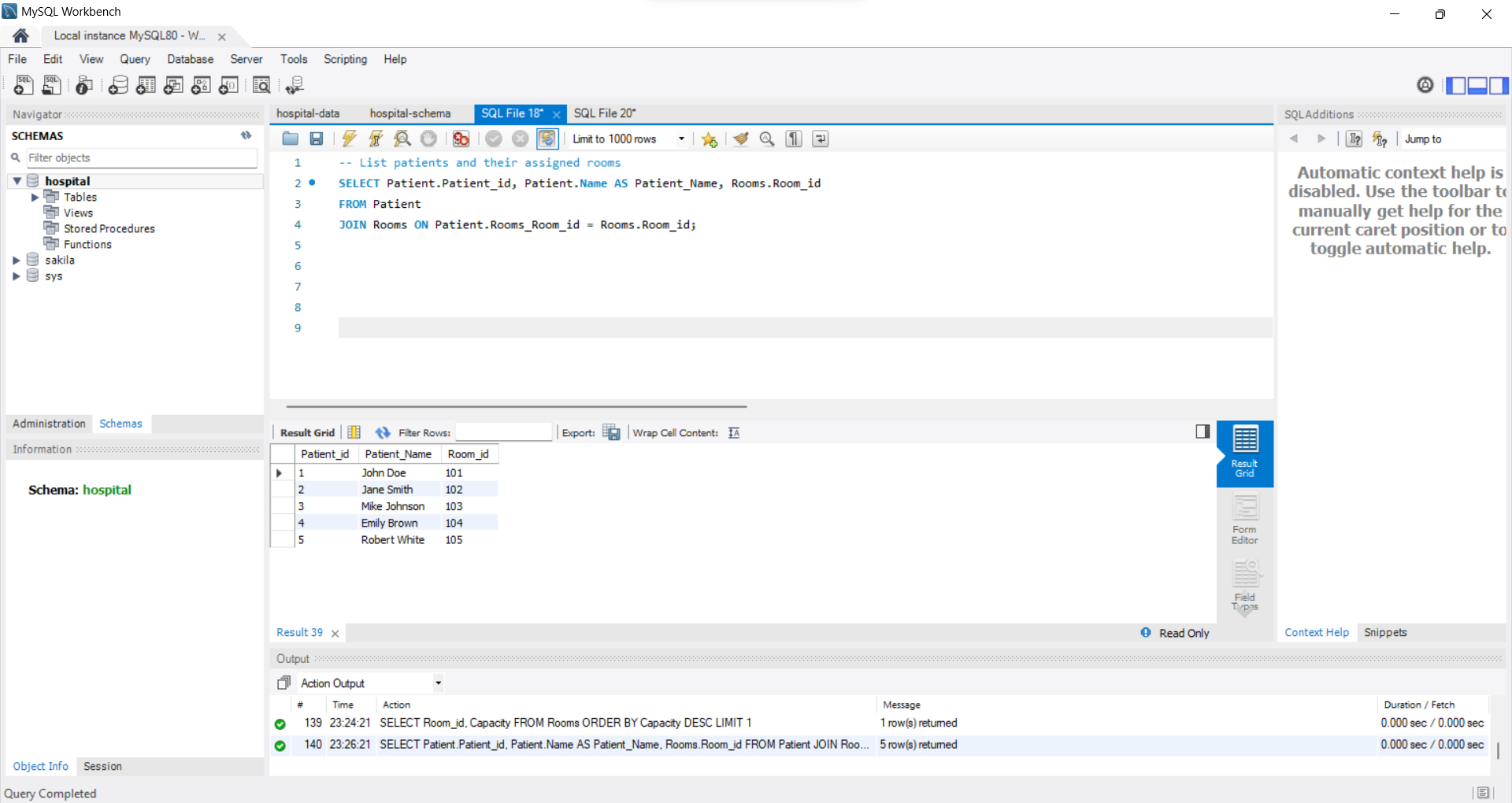












7- Transactions and description

AdmitPatientTransaction:

Description: A transaction to admit a new patient and record their health status simultaneously.

Body: A START TRANSACTION block with multiple SQL statements to insert into the Patient and HealthRecord tables.

Usefulness Discussion: Ensures the atomicity of the operation, preventing inconsistencies if either part of the transaction fails.

DischargePatientTransaction:

Description: A transaction to discharge a patient and update room occupancy.

Body: A START TRANSACTION block with SQL statements to delete from the Patient table and update the RoomOccupancyView.

Usefulness Discussion: Guarantees that both the patient deletion and room occupancy update occur together, maintaining data integrity.