Hospital DB Phase 3

Surya Vadivazhagu & James Flynn Team 37

We recorded a session of running the SQL part as well as running the Java file for Part 3 to prove it ran on the CCC machines using the *script* UNIX command. In the SQL file we write how many results should come up based on the insert statements we wrote for the views. Please consider them when examining the views to ensure the data matches up.

Requirements for Part 2 are at the end of the PDF.

SQL:

Script started on Sun Feb 23 20:49:36 2020

bash-4.2$ sqlplus svadivazhagu@CS

SQL\*Plus: Release 18.0.0.0.0 - Production on Sun Feb 23 20:49:40 2020

Version 18.5.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Enter password:

Last Successful login time: Sun Feb 23 2020 20:47:58 -05:00

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> @phase3.sql

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table dropped.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

View created.

View created.

no rows selected

no rows selected

no rows selected

Trigger created.

Trigger created.

Trigger created.

Trigger created.

Trigger created.

Trigger created.

Trigger created.

Trigger created.

SQL> exit

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

bash-4.2$ exit

exit

Script done on Sun Feb 23 20:50:02 2020

Java:

Script started on Mon Feb 24 12:35:40 2020

[?1034hbash-4.2$ javac REp  eporting.java

bash-4.2$ java Reportings  svadivazhagu 321

Connecting to the database...

Logged in

1- Report Patients Basic Information

2- Report Doctors Basic Information

3- Report Admissions Information

4- Update Admissions Payment

Disconnected from the server.

bash-4.2$ cl  java re  Reporting svadiv     vadivazhagu 321 1

Connecting to the database...

Logged in

Enter Patient SSN: 100081645

 Now Running SELECT \* FROM Patient WHERE SSN = '100081645'

Patient SSN: 100081645

Patient First Name: William

Patient Last Name: Jacobs

Patient Address: 07809 Sanchez Falls Apt.

Disconnected from the server.

bash-4.2$ clear     java Reporting svadivazhagu 321 1 2

Connecting to the database...

Logged in

Enter Doctor ID: 59835

 Now Running SELECT \* FROM Doctor WHERE Id = '59835'

Doctor ID: 59835

Doctor First Name: Amanda

Doctor Last Name: Green

Doctor Gender: F

Disconnected from the server.

bash-4.2$ java Reporting svadivazhagu 321 2 3

Connecting to the database...

Logged in

Enter Admission Number: 1

 Now Running SELECT \* FROM Admission WHERE AdmissionNum = 1

 Now Running SELECT DISTINCT RoomNum, StartDate, EndDate FROM StayIn WHERE AdmissionNum = 1

 Now Running SELECT DISTINCT DoctorId FROM Examine WHERE AdmissionNum = 1

Admission Number: 1

Admission Date: 2019-01-02 00:00:00

Patient SSN: 100081645

TotalPayment: 500000

Rooms:

    RoomNum: 172  From Date: 2019-01-02 00:00:00  End Date: 2019-02-15 00:00:00

Doctors examined the patient in this admission:

    Doctor ID: 59835

java.sql.SQLException: Closed Resultset: next

    at oracle.jdbc.driver.InsensitiveScrollableResultSet.ensureOpen(InsensitiveScrollableResultSet.java:114)

    at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:402)

    at Reporting.queryRun(Reporting.java:66)

    at Reporting.main(Reporting.java:164)

Disconnected from the server.

bash-4.2$ java Reporting svadivazhagu 321 3 4

Connecting to the database...

Logged in

Enter Admission Number: 1

Enter the new total payment: 1000000000          9999

Now Running UPDATE Admission SET TotalPayment = 9999 WHERE AdmissionNum= 1

Successfully updated admission number 1's total payment to $9999.

Disconnected from the server.

bash-4.2$ java Reporting svadivazhagu 321 4 3

Connecting to the database...

Logged in

Enter Admission Number: 1

 Now Running SELECT \* FROM Admission WHERE AdmissionNum = 1

 Now Running SELECT DISTINCT RoomNum, StartDate, EndDate FROM StayIn WHERE AdmissionNum = 1

 Now Running SELECT DISTINCT DoctorId FROM Examine WHERE AdmissionNum = 1

Admission Number: 1

Admission Date: 2019-01-02 00:00:00

Patient SSN: 100081645

TotalPayment: 9999

Rooms:

    RoomNum: 172  From Date: 2019-01-02 00:00:00  End Date: 2019-02-15 00:00:00

Doctors examined the patient in this admission:

    Doctor ID: 59835

java.sql.SQLException: Closed Resultset: next

    at oracle.jdbc.driver.InsensitiveScrollableResultSet.ensureOpen(InsensitiveScrollableResultSet.java:114)

    at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:402)

    at Reporting.queryRun(Reporting.java:66)

    at Reporting.main(Reporting.java:164)

Disconnected from the server.

bash-4.2$ exit

exit

Script done on Mon Feb 24 12:37:34 2020

There is a SQL Exception that is sometimes caught but it doesn’t impact the actual functionality and the program runs as it should. All the functionality works as it should.

Requirements for Part 2:

If a doctor visits a patient in the ICU, they must leave a comment.

Event Type: Before

Operation: Insert

Granularity: For each row

Table: Examine

• The insurance payment should be calculated automatically as 65% of the total

payment. If the total payment changes then the insurance amount should also

change.

Event Type: Before

Operation: Insert or Update

Granularity: for each row

Table: Admission

• Ensure that regular employees (with rank 0) must have their supervisors as

division managers (with rank 1). Also each regular employee must have a

supervisor at all times.

Event Type: Before

Operation: Insert or Update

Granularity: for each row

Table: Employee

• Similarly, division managers (with rank 1) must have their supervisors as general managers (with rank 2). Division managers must have supervisors at all times.

Event Type: Before

Operation: Insert or Update

Granularity: for each row

Table: Employee

General Managers must not have any supervisors.

Event Type: Before

Operation: Insert or Update

Granularity: for each row

Table: Employee

• When a patient is admitted to an Emergency Room (a room with an Emergency

service) on date D, the futureVisitDate should be automatically set to 2 months

after that date, i.e., D + 2 months. The futureVisitDate may be manually changed

later, but when the Emergency Room admission happens, the date should be set

to default as mentioned above.

Event Type: Before

Operation: Insert

Granularity: For each row

Table: StayIn

• If a piece of equipment is of type ‘CT Scanner’ or ‘Ultrasound’, then the purchase year must be not null and after 2006.

Event Type: before

Operation: insert or update

Granularity: for each row

Table: Equipment

• When a patient leaves the hospital (Admission leave time is set), print out the

patient’s first and last name, address, all of the comments from doctors involved

in that admission, and which doctor (name) left each comment.

Event Type: after

Operation: update

Granularity: for each row

Table: admission