CHAPTER 5

Advanced SQL

Solutions for the Practice Exercises of Chapter 5

Practice Exercises

5.1

Answer:
It prints out the manager of "dog," that manager's manager, etc., until we reach a manager who has no manager (presumably, the CEO, who most certainly is a cat). Note: If you try to run this, use your own Oracle ID and password.

5.2

Answer:
Please see Figure 5.101

5.3

Answer:
Please see Figure 5.102

5.4

Answer:
Writing queries in SQL is typically much easier than coding the same queries in a general-purpose programming language. However, not all kinds of queries can be written in SQL. Also, nondeclarative actions such as printing a report, interacting with a user, or sending the results of a query to a graphical user interface cannot be done from within SQL. Under circumstances in which we want the best of both worlds, we can choose embedded SQL or dynamic SQL, rather than using SQL alone or using only a general-purpose programming language.

5.5

Answer:

```
printTable(ResultSet result) throws SQLException {
    metadata = result.getMetaData();
    num_cols = metadata.getColumnCount();
    for(int i = 1; i <= num_cols; i++) {
        System.out.print(metadata.getColumnName(i) + '\t');
    }
    System.out.println();
    while(result.next()) {
        for(int i = 1; i <= num_cols; i++) {
            System.out.print(result.getString(i) + '\t'
        }
        System.out.println();
    } }
```

Figure 5.101 Java method using JDBC for ??.

Please see Figure 5.103

5.6

Answer:
Please see Figure 5.104

5.7

Answer:

create trigger *check-delete-trigger* **after delete on** *account*
referencing old row as *orow*
for each row
delete from *depositor*
where *depositor.customer_name* **not in**
(**select** *customer_name* **from** *depositor*
where *account_number* <> *orow.account_number*)
end

5.8

Answer:

```
import java.sql.*;
import java.util.Scanner;
import java.util.Arrays;
public class AllCoursePrereqs {
  public static void main(String[] args) {
    try (
         Connection con=DriverManager.getConnection
           ("jdbc:oracle:thin:@edgar0.cse.lehigh.edu:1521:cse241","star","pw");
         Statement s=con.createStatement();
         ){
           String q;
           String c;
           ResultSet result;
           int maxCourse = 0;
           q = "select count(*) as C from course";
           result = s.executeQuery(q);
           if (!result.next()) System.out.println ("Unexpected empty result.");
           else maxCourse = Integer.parseInt(result.getString("C"));
           int numCourse = 0, oldNumCourse = -1;
           String[] prereqs = new String [maxCourse];
           Scanner krb = new Scanner(System.in);
           System.out.print("Input a course id (number): ");
           String course = krb.next();
           String courseString = "" + '\'' + course + '\'';
           while (numCourse != oldNumCourse) {
             for (int i = oldNumCourse + 1; i < numCourse; i++) {
               courseString += ", " + '\'' + prereqs[i] + '\'' ;
             }
             oldNumCourse = numCourse;
             q = "select prereq_id from prereq  where course_id in ("
               + courseString + ")";
             result = s.executeQuery(q);
             while (result.next()) {
               c = result.getString("prereq_id");
               boolean found = false;
               for (int i = 0; i < numCourse; i++)
                    found |= prereqs[i].equals(c);
               if (!found) prereqs[numCourse++] = c;
             }
             courseString = "" + '\'' + prereqs[oldNumCourse] + '\'';
           }
           Arrays.sort(prereqs,0,numCourse);
           System.out.print("The courses that must be taken prior to "
             + course + " are: ");
           for (int i = 0; i < numCourse; i++)
                System.out.print ((i==0?" ":", ") + prereqs[i]);
           System.out.println();
         } catch(Exception e){e.printStackTrace();
} }
```

Figure 5.102 Complete Java program using JDBC for ??.

```
create trigger onesec before insert on section
referencing new row as nrow
for each row
when (nrow.time_slot_id in (
   select time_slot_id
   from teaches natural join section
   where ID in (
            select ID
            from teaches natural join section
            where sec_id = nrow.sec_id and course_id = nrow.course_id and
                  semester = nrow.semester and year = nrow.year
)))
begin
   rollback
end;


create trigger oneteach before insert on teaches
referencing new row as nrow
for each row
when (exists (
            select time_slot_id
            from teaches natural join section
            where ID = nrow.ID
   intersect
            select time_slot_id
            from section
            where sec_id = nrow.sec_id and course_id = nrow.course_id and
                  semester = nrow.semester and year = nrow.year
))
begin
   rollback
end;
```

Figure 5.103 Trigger code for ??.

```
select *
from (
        select student, total, rank() over (order by (total) desc) as t_rank
        from (
                select student, sum(marks) as total
                from S group by student
        )
)
where t_rank <= 10
```

5.9

Answer:

```
select year, month, day, shares_traded,
        rank( ) over (order by shares_traded desc ) as mostshares
from nyse
```

5.10

Answer:

```
select year, month, day, sum(shares_traded) as shares,
        sum(num_trades) as trades, sum(dollar_volume) as total_volume
from nyse
group by rollup (year, month, day)
```

5.11

Answer:

```
groupby rollup(a), rollup(b), rollup(c ), rollup(d)
```

```
create trigger insert_into_branch_cust_via_depositor
after insert on depositor
referencing new row as inserted
for each row
insert into branch_cust
      select branch_name, inserted.customer_name
      from account
      where inserted.account_number = account.account_number

create trigger insert_into_branch_cust_via_account
after insert on account
referencing new row as inserted
for each statement
insert into branch_cust
      select inserted.branch_name, customer_name
      from depositor
      where depositor.account_number = inserted.account_number
```

Figure 5.104 Trigger code for ??.