Graded Problems for Module 5

The graded problems in module 5 involve SELECT statements for multiple tables and row summaries and data manipulation statements using the INSERT, UPDATE, and DELETE statements of SQL. You should execute the statements using either Oracle, MySQL, or PostgreSQL database server. The last set of problems involves rewriting SQL statements with errors and poor formatting.

To facilitate grading, please number the SQL statements and format them neatly. You do not need to show the result tables. Indicate in the beginning of your document if you used Oracle, MySQL, or PostgreSQL.

SELECT Statement Problems

1. For event requests, list the event number, event date (*eventrequest.dateheld*), and count of the event plans. Only include event requests in the result if the event request has more than one related event plan with a work date in December 2018.

```
SELECT eventrequest.eventno, eventrequest.dateheld, count(*) AS EventPlanCount
FROM eventrequest
INNER JOIN eventplan
ON eventrequest.eventno = eventplan.eventno
WHERE (eventrequest.dateheld BETWEEN '01-Dec-2018' AND '31-Dec-2018')
GROUP BY eventrequest.eventno, eventrequest.dateheld
HAVING count(*) > 1;
```

2. List the plan number, event number, work date, and activity of event plans meeting the following two conditions: (1) the work date is in December 2018 and (2) the event is held in the “Basketball arena”. Your query must not use the facility number (“F101”) of the basketball arena in the WHERE clause. Instead, you should use a condition on the *FacName* column for the value of “Basketball arena”.

```
SELECT eventplan.planno, eventrequest.eventno, eventplan.workdate, eventplan.activity
FROM eventplan
INNER JOIN eventrequest
ON eventrequest.eventno = eventplan.eventno
INNER JOIN facility
ON eventrequest.facno = facility.facno
WHERE (eventplan.workdate BETWEEN '01-Dec-2018' AND '31-Dec-2018') AND
facility.facname = 'Basketball arena';
```

3. List the event number, event date, status, and estimated cost of events where there is an event plan managed by Mary Manager and the event is held in the basketball arena in the period October 1 to December 31, 2018. Your query must not use the facility number ("F101") of the basketball arena or the employee number ("E101") of "Mary Manager" in the WHERE clause. Thus, the WHERE clause should not have conditions involving the facility number or employee number compared to constant values.

```
SELECT eventrequest.eventno, eventrequest.dateheld, eventrequest.status, eventrequest.estcost
FROM eventplan
INNER JOIN eventrequest
ON eventrequest.eventno = eventplan.eventno
INNER JOIN facility
ON eventrequest.facno = facility.facno
INNER JOIN employee
ON eventplan.empno = employee.empno
```

```
WHERE employee.empname = 'Mary Manager' AND facility.facname = 'Basketball arena'
AND (eventrequest.dateheld BETWEEN '01-Oct-2018' AND '31-Dec-2018');
```

4. List the plan number, line number, resource name, number of resources (*eventplanline.number*), location name, time start, and time end where the event is held at the basketball arena, the event plan has activity of activity of “Operation”, and the event plan has a work date in the period October 1 to December 31, 2018. Your query must not use the facility number (“F101”) of the basketball arena in the WHERE clause. Instead, you should use a condition on the *FacName* column for the value of “Basketball arena”.

```
SELECT eventplanline.planno, eventplanline.lineno, resourcetbl.resname,
eventplanline.numberfld,
location.locname, eventplanline.timestart, eventplanline.timeend
FROM eventplanline
INNER JOIN resourcetbl
ON eventplanline.resno = resourcetbl.resno
INNER JOIN location
ON eventplanline.locno = location.locno
INNER JOIN eventplan
ON eventplanline.planno = eventplan.planno
INNER JOIN facility
ON location.facno = facility.facno
WHERE (facility.facname = 'Basketball arena')
AND (eventplan.activity = 'Operation')
AND (eventplan.workdate BETWEEN '01-Oct-2018' AND '31-Dec-2018');
```

Database Modification Problems

1. Insert a new row into the *Facility* table with facility name “Swimming Pool”.

```
INSERT INTO facility
(facno, facname)
VALUES ('F104','Swimming Pool');
```

2. Insert a new row in the *Location* table related to the *Facility* row in modification problem 1. The new row should have “Door” for the location name.

```
INSERT INTO location
(locno, facno, locname)
VALUES ('L107', 'F104', 'Door');
```

3. Insert a new row in the *Location* table related to the *Facility* row in modification problem 1. The new row should have “Locker Room” for the location name.

```
INSERT INTO location
(locno, facno, locname)
VALUES ('L108', 'F104', 'Locker Room');
```

4. Change the location name of “Door” to “Gate” for the row inserted in modification problem 2. In MySQL, you need to place the UPDATE statement between two SET statements.

```
SET SQL_SAFE_UPDATES = 0;
```

UPDATE statement

SET SQL_SAFE_UPDATES = 1;

UPDATE location

SET locname = 'Gate'

WHERE locno = 'L107'

AND facno = 'F104';

5. Delete the row inserted in modification problem 3.

DELETE FROM location

WHERE locno = 'L108'

AND facno = 'F104';

SQL Statements with Errors and Poor Formatting

1. Identify errors in the following SQL statement and label errors with error type (syntax, redundancy, or semantic). To simplify your work, the statement has only one type of error. Rewrite the statement to remove the error.

```
SELECT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, employee, facility, eventplan
WHERE estaudience > 5000
  AND eventplan.empno = employee.empno
  AND eventrequest.facno = facility.facno
  AND facname = 'Football stadium'
  AND empname = 'Mary Manager'
```

```
SEMANTIC
SELECT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, employee, facility, eventplan
WHERE estaudience > 5000
  AND eventplan.empno = employee.empno
  AND eventrequest.facno = facility.facno
```

```
  AND eventplan.eventno = eventrequest.eventno
  AND facname = 'Football stadium'
  AND empname = 'Mary Manager'
```

2. Identify errors in the following SQL statement and label errors with error type (syntax, redundancy, or semantic). To simplify your work, the statement has only one type of error. Rewrite the statement to remove the error.

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, eventplan
WHERE estaudience > 4000
  AND eventplan.eventno = eventrequest.eventno
GROUP BY eventrequest.eventno, dateheld, status, estcost
```

REDUNDANCY

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, eventplan
WHERE estaudience > 4000
  AND eventplan.eventno = eventrequest.eventno
~~GROUP BY eventrequest.eventno, dateheld, status, estcost~~
```

3. Identify errors in the following SQL statement and label errors with error type (syntax, redundancy, or semantic). To simplify your work, the statement has only one type of error. Rewrite the statement to remove the error.

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, employee, facility, eventplan
WHERE estaudience > 5000
  AND eventplan.empno = employee.empno
  AND eventrequest.facno = facility.facno
  AND eventplan.eventno = eventrequest.eventno
  AND facname = 'Football stadium'
```

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, ~~employee,~~ facility, eventplan
WHERE estaudience > 5000
  ~~AND eventplan.empno = employee.empno~~
  AND eventrequest.facno = facility.facno
  AND eventplan.eventno = eventrequest.eventno
  AND facname = 'Football stadium'
```

REDUNDANCY

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, facility, eventplan
WHERE estaudience > 5000
  AND eventrequest.facno = facility.facno
  AND eventplan.eventno = eventrequest.eventno
  AND facname = 'Football stadium'
```

4. Identify errors in the following SQL statement and label errors with error type (syntax, redundancy, or semantic). To simplify your work, the statement has only one type of error. Rewrite the statement to remove the errors.

```
SELECT DISTINCT eventno, dateheld, status, estcost
FROM eventrequest, employee, eventplan
WHERE estaudience BETWEEN 5000 AND 10000
  AND eventplan.empno = employee.empno
  AND eventrequest.eventno = eventplan.eventno
  AND empname = 'Mary Manager'
```

SYNTAX

```
SELECT DISTINCT eventrequest.eventno, dateheld, status, estcost
FROM eventrequest, employee, eventplan
WHERE estaudience BETWEEN 5000 AND 10000
  AND eventplan.empno = employee.empno
  AND eventrequest.eventno = eventplan.eventno
  AND empname = 'Mary Manager'
```

5. Identify areas in which the SQL statement has poor coding practices and rewrite the statement to improve the coding practices. You do not need to search for errors.

```
      SELECT eventplan.planno, lineno, resname,
numberfld, timestart, timeend
    FROM eventrequest, facility, eventplan,
eventplanline, resourcetbl
     WHERE estaudience = '10000'
AND eventplan.planno =
   eventplanline.planno AND eventrequest.facno
   = facility.facno
         AND facname =
```

```
'Basketball arena' AND
   eventplanline.resno = resourcetbl.resno
      AND eventrequest.eventno = eventplan.eventno
```

SELECT eventplan.planno, lineno, resname, numberfld, timestart, timeend

FROM eventrequest, facility, eventplan, eventplanline, resourcetbl

WHERE estaudience = '10000'

AND eventplan.planno = eventplanline.planno

AND eventrequest.facno = facility.facno

AND eventplanline.resno = resourcetbl.resno

AND eventrequest.eventno = eventplan.eventno

AND facname = 'Basketball arena';