

Module 5 Problem Solutions

The problems use the intercollegiate athletics database as described in the Order Entry Database Background document. The course website also contains CREATE TABLE and INSERT statements for Oracle, MySQL, and PostgreSQL.

1. List the event number, date held, customer number, customer name, facility number, and facility name of 2018 events placed by Boulder customers.
2. List the customer number, customer name, event number, date held, facility number, facility name, and estimated audience cost per person ($\text{EstCost} / \text{EstAudience}$) for events held on 2018, in which the estimated cost per person is less than \$0.2
3. List the customer number, customer name, and total estimated costs for Approved events.
The total amount of events is the sum of the estimated cost for each event. Group the results by customer number and customer name.
4. Insert yourself as a new row in the *Customer* table.
5. Increase the rate by 10 percent of nurse resource in the resource table.
6. Delete the new row added to the *Customer* table.

Solutions

```
1.
Oracle SQL solutions
SELECT EventNo, DateHeld, Customer.CustNo, CustName,
       Facility.FacNo, FacName
FROM EventRequest, Customer, Facility
WHERE City = 'Boulder'
      AND DateHeld BETWEEN '1-Jan-2018' AND '31-Dec-2018'
      AND EventRequest.CustNo = Customer.CustNo
      AND EventRequest.FacNo = Facility.FacNo;

SELECT EventNo, DateHeld, Customer.CustNo, CustName,
       Facility.FacNo, FacName
FROM EventRequest INNER JOIN Customer
```

```
ON EventRequest.CustNo = Customer.CustNo
INNER JOIN Facility ON EventRequest.FacNo = Facility.FacNo
WHERE City = 'Boulder'
AND DateHeld BETWEEN '1-Jan-2018' AND '31-Dec-2018';
```

MySQL and PostgreSQL solutions

```
SELECT EventNo, DateHeld, Customer.CustNo, CustName,
       Facility.FacNo, FacName
FROM EventRequest, Customer, Facility
WHERE City = 'Boulder'
AND DateHeld BETWEEN '2018-01-01' AND '2018-12-31'
AND EventRequest.CustNo = Customer.CustNo
AND EventRequest.FacNo = Facility.FacNo;
```

```
SELECT EventNo, DateHeld, Customer.CustNo, CustName,
       Facility.FacNo, FacName
FROM EventRequest INNER JOIN Customer
ON EventRequest.CustNo = Customer.CustNo
INNER JOIN Facility ON EventRequest.FacNo = Facility.FacNo
WHERE City = 'Boulder'
AND DateHeld BETWEEN '2018-01-01' AND '2018-12-31';
```

2.

Oracle solutions

```
SELECT Customer.CustNo, CustName, EventNo,
       DateHeld, Facility.FacNo, FacName, EstCost/EstAudience AS AudCost
FROM EventRequest, Customer, Facility
WHERE DateHeld BETWEEN '1-Jan-2018' AND '31-Dec-2018'
AND EstCost/EstAudience < 0.2
AND EventRequest.CustNo = Customer.CustNo
AND EventRequest.FacNo = Facility.FacNo;
```

```
SELECT Customer.CustNo, CustName, EventNo,
       DateHeld, Facility.FacNo, FacName, EstCost/EstAudience AS AudCost
FROM EventRequest INNER JOIN Customer
ON EventRequest.CustNo = Customer.CustNo
INNER JOIN Facility ON EventRequest.FacNo = Facility.FacNo
WHERE DateHeld BETWEEN '1-Jan-2018' AND '31-Dec-2018'
AND EstCost/EstAudience < 0.2;
```

MySQL and PostgreSQL solutions

```
SELECT Customer.CustNo, CustName, EventNo,
       DateHeld, Facility.FacNo, FacName, EstCost/EstAudience AS AudCost
FROM EventRequest, Customer, Facility
WHERE DateHeld BETWEEN '2018-01-01' AND '2018-12-31'
```

```
AND EstCost/EstAudience < 0.2  
AND EventRequest.CustNo = Customer.CustNo  
AND EventRequest.FacNo = Facility.FacNo;
```

```
SELECT Customer.CustNo, CustName, EventNo,  
       DateHeld, Facility.FacNo, FacName, EstCost/EstAudience AS AudCost  
FROM EventRequest INNER JOIN Customer  
     ON EventRequest.CustNo = Customer.CustNo  
     INNER JOIN Facility ON EventRequest.FacNo = Facility.FacNo  
WHERE DateHeld BETWEEN '2018-01-01' AND '2018-12-31'  
     AND EstCost/EstAudience < 0.2;
```

3.

Oracle, MySQL, and PostgreSQL solutions

```
SELECT Customer.CustNo, CustName, SUM(EstCost) AS TotEstCost  
FROM EventRequest, Customer  
WHERE Status = 'Approved'  
     AND EventRequest.CustNo = Customer.CustNo  
GROUP BY Customer.CustNo, CustName;
```

```
SELECT Customer.CustNo, CustName, SUM(EstCost) AS TotEstCost  
FROM EventRequest INNER JOIN Customer  
     ON EventRequest.CustNo = Customer.CustNo  
WHERE Status = 'Approved'  
GROUP BY Customer.CustNo, CustName;
```

4.

```
INSERT INTO Customer  
(CustNo, CustName, Address, Internal, Contact, Phone, City, State, Zip)  
VALUES ('C9999999', 'Michael Mannino', '123 Any Street', 'Y', 'Self', '720000',  
       'Denver', 'CO', '80204');
```

5.

Oracle and PostgreSQL

```
UPDATE Resourcetbl  
SET Rate = Rate * 1.1  
WHERE ResName = 'nurse';
```

MySQL solution

UPDATE and DELETE statements will not execute if WHERE conditions do not reference the primary key. The SQL_SAFE_UPDATES option set to 0 allows UPDATE and DELETE statements without a WHERE condition on the primary key.

```
SET SQL_SAFE_UPDATES = 0;
```

```
UPDATE Resourcetbl  
  SET Rate = Rate * 1.1  
  WHERE ResName = 'nurse';
```

```
SET SQL_SAFE_UPDATES = 1;
```

6.

```
DELETE FROM Customer  
  WHERE CustNo = 'C9999999';
```