

Module 4 Problem Solutions

The problems use the intercollegiate athletics database. The course website also contains Oracle and PostgreSQL CREATE TABLE statements and INSERT statements.

1. List the customer number, the name, the phone number, and the city of customers.
2. List the customer number, the name, the phone number, and the city of customers who reside in Colorado (State is CO).
3. List all columns of the *EventRequest* table for events costing more than \$4000. Order the result by the event date (*DateHeld*).
4. List the event number, the event date (*DateHeld*), and the estimated audience number with approved status and audience greater than 9000 or with pending status and audience greater than 7000.
5. List the event number, event date (*DateHeld*), customer number and customer name of events placed in January 2022 by customers from Boulder.
6. List the average number of resources used (*NumberFld*) by plan number. Include only location number L100.
7. List the average number of resources used (*NumberFld*) by plan number. Only include location number L100. Eliminate plans with less than two event lines containing location number L100.

Solutions

All statements execute in both Oracle and PostgreSQL except where noted.

1.

```
SELECT CustNo, CustName, Phone, City  
FROM Customer;
```
- 2.

```
SELECT CustNo, CustName, Phone, City
FROM Customer
WHERE State = 'CO';
```

```
3.
SELECT *
FROM EventRequest
WHERE EstCost > 4000
ORDER BY DateHeld;
```

4. The parentheses are necessary when mixing the logical AND and OR connectors.

```
SELECT EventNo, DateHeld, Status, EstAudience
FROM EventRequest
WHERE (Status = 'Approved' AND EstAudience > 9000)
      OR (Status = 'Pending' AND EstAudience > 7000);
```

```
5.
SELECT EventNo, DateHeld, Customer.CustNo, CustName
FROM EventRequest, Customer
WHERE City = 'Boulder'
      AND DateHeld BETWEEN '1-Dec-2022' AND '31-Dec-2022'
      AND EventRequest.CustNo = Customer.CustNo;
```

```
SELECT EventNo, DateHeld, Customer.CustNo, CustName
FROM EventRequest INNER JOIN Customer
  ON EventRequest.CustNo = Customer.CustNo
WHERE City = 'Boulder'
      AND DateHeld BETWEEN '1-Dec-2022' AND '31-Dec-2022' ;
```

Alternative date formats for PostgreSQL

```
SELECT EventNo, DateHeld, Customer.CustNo, CustName
FROM EventRequest, Customer
WHERE City = 'Boulder'
      AND DateHeld BETWEEN '2022-12-01' AND '2022-12-31'
      AND EventRequest.CustNo = Customer.CustNo;
```

```
SELECT EventNo, DateHeld, Customer.CustNo, CustName
FROM EventRequest INNER JOIN Customer
  ON EventRequest.CustNo = Customer.CustNo
WHERE City = 'Boulder'
      AND DateHeld BETWEEN '2022-12-01' AND '2022-12-31';
```

6.

```
SELECT PlanNo, AVG(NumberFld) AS AvgNumResources
FROM EventPlanLine
WHERE LocNo = 'L100'
GROUP BY PlanNo;
```

7.

```
SELECT PlanNo, AVG(NumberFld) AS AvgNumResources,
       COUNT(*) AS NumEventLines
FROM EventPlanLine
WHERE LocNo = 'L100'
GROUP BY PlanNo
HAVING COUNT(*) > 1;
```