Module-4

 # List the city, state, and zip codes in the customer table. Your result should not have duplicates. (Hint: The DISTINCT keyword eliminates duplicates.)

```
SELECT DISTINCT
  c.City,
  c.State,
  c.Zip
FROM Customer AS c ;
```

```
mysql> SELECT DISTINCT
    ->   c.City,
    ->   c.State,
    ->   c.Zip
    -> FROM Customer AS c ;
+------------+-------+-------+
| City       | State | Zip   |
+------------+-------+-------+
| Boulder    | CO    | 80309 |
| Louisville | CO    | 80027 |
+------------+-------+-------+
2 rows in set (0.00 sec)
```

 # List the name, department, phone number, and email address of employees with a phone number beginning with “3-”.

```
SELECT
  e.EmpName,
  e.Department,
  e.Phone,
  e.Email
FROM Employee AS e
WHERE e.phone LIKE '3-%';
```

```
mysql> SELECT
    ->   e.EmpName,
    ->   e.Department,
    ->   e.Phone,
    ->   e.Email
    -> FROM Employee AS e
    -> WHERE e.phone LIKE '3-%';
+--------------------+----------------+--------+--------------------+
| EmpName            | Department     | Phone  | Email              |
+--------------------+----------------+--------+--------------------+
| Chuck Coordinator  | Administration | 3-1111 | chuck@colorado.edu |
| Sally Supervisor   | Planning       | 3-2222 | sally@colorado.edu |
| Alan Administrator | Administration | 3-3333 | alan@colorado.edu  |
+--------------------+----------------+--------+--------------------+
3 rows in set (0.00 sec)
```

 # List all columns of the resource table with a rate between $10 and $20. Sort the result by rate.

```
SELECT *
FROM ResourceTbl AS r
WHERE r.Rate >= 10 AND r.Rate <= 20
ORDER BY r.Rate;
```

```
mysql> SELECT *
    -> FROM ResourceTbl AS r
    -> WHERE r.Rate >= 10 AND r.Rate <= 20
    -> ORDER BY r.Rate;
+-------+--------------+-------+
| ResNo | ResName      | Rate  |
+-------+--------------+-------+
| R100  | attendant    | 10.00 |
| R102  | usher        | 10.00 |
| R105  | food service | 10.00 |
| R101  | police       | 15.00 |
| R104  | janitor      | 15.00 |
| R103  | nurse        | 20.00 |
+-------+--------------+-------+
6 rows in set (0.00 sec)
```

 # List the event requests with a status of “Approved” or “Denied” and an authorized date in July 2013. Include the event number, authorization date, and status in the output. (Hint: see the examples in Module 4 for date constants in Oracle and MySQL.)

```
SELECT
  er.EventNo,
  er.DateAuth,
  er.Status
FROM EventRequest AS er
WHERE (er.Status = 'Approved' OR er.Status = 'Denied') AND er.DateAuth BETWEEN
'2013-07-01' AND '2013-07-31';
```

```
mysql> SELECT
    ->   er.EventNo,
    ->   er.DateAuth,
    ->   er.Status
    -> FROM EventRequest AS er
    -> WHERE (er.Status = 'Approved' OR er.Status = 'Denied') AND er.DateAuth BETWEEN '2013-07-01' AND '2013-07-31';
+---------+------------+----------+
| EventNo | DateAuth   | Status   |
+---------+------------+----------+
| E102    | 2013-07-31 | Approved |
| E104    | 2013-07-31 | Approved |
| E106    | 2013-07-31 | Approved |
| E107    | 2013-07-31 | Denied   |
+---------+------------+----------+
4 rows in set (0.00 sec)
```

 # List the location number and name of locations that are part of the “Basketball arena”. Your WHERE clause should not have a condition involving the facility number compared to a constant (“F101”). Instead, you should use a condition on the FacName column for the value of “Basketball arena”.

```
SELECT
  l.LocNo,
  l.LocName
FROM Location AS l, Facility AS f
WHERE l.FacNo = f.FacNo AND f.FacName = 'Basketball arena';
```

```
mysql> SELECT
    ->   l.LocNo,
    ->   l.LocName
    -> FROM Location AS l, Facility AS f
    -> WHERE l.FacNo = f.FacNo AND f.FacName = 'Basketball arena';
+-------+-------------+
| LocNo | LocName     |
+-------+-------------+
| L103  | Locker room |
| L105  | Gate        |
+-------+-------------+
2 rows in set (0.00 sec)
```

 # For each event plan, list the plan number, count of the event plan lines, and sum of the number of resources assigned. For example, plan number "P100" has 4 lines and 7 resources assigned. You only need to consider event plans that have at least one line.

```
SELECT
  ep.PlanNo,
  count(el.PlanNo),
  sum(el.NumberFld)
FROM EventPlan AS ep, EventPlanLine AS el
WHERE ep.PlanNo = el.PlanNo
GROUP BY ep.PlanNo;
```

```
mysql> SELECT
    ->   ep.PlanNo,
    ->   count(el.PlanNo),
    ->   sum(el.NumberFld)
    -> FROM EventPlan AS ep, EventPlanLine AS el
    -> WHERE ep.PlanNo = el.PlanNo
    -> GROUP BY ep.PlanNo;
+--------+------------------+--------------------+
| PlanNo | count(el.PlanNo) | sum(el.NumberFld) |
+--------+------------------+--------------------+
| P100   |                4 |                  6 |
| P102   |                3 |                  8 |
| P103   |                3 |                  8 |
| P104   |                2 |                  8 |
| P105   |                2 |                  8 |
| P199   |                1 |                  1 |
| P349   |                1 |                  1 |
| P85    |                3 |                 10 |
| P95    |                5 |                 16 |
+--------+------------------+--------------------+
9 rows in set (0.00 sec)
```