Practice Exercise #1.1: (create)

Q1: Create the table suppliers that contains four columns: supplier_id, supplier_name, city, state and supplier_id is primary key

Solution:

```
CREATE TABLE suppliers
( supplier_id int NOT NULL,
  supplier_name char(50) NOT NULL,
  city char(50),
  state char(25),
  CONSTRAINT suppliers_pk PRIMARY KEY (supplier_id)
);
```

Practice Exercise #1.2 (insert)

Q2: Insert some records in the table suppliers that contains four columns: supplier_id, supplier_name, city, state and supplier_id is primary key

Solution:

```
INSERT INTO suppliers     (supplier_id, supplier_name, city, state)
VALUES   (100, 'Microsoft', 'Redmond', 'Washington'),
         (200, 'Google', 'Mountain View', 'California'),
         (300, 'Oracle', 'Redwood City', 'California'),
         (400, 'Kimberly-Clark', 'Irving', 'Texas'),
         (500, 'Tyson Foods', 'Springdale', 'Arkansas'),
         (600, 'SC Johnson', 'Racine', 'Wisconsin'),
         (700, 'Dole Food Company', 'Westlake Village', 'California'),
         (800, 'Flowers Foods', 'Thomasville', 'Georgia'),
         (900, 'Electronic Arts', 'Redwood City', 'California');
```

Practice Exercise #1.3: (select)

Q3: Based on the supplier table, select the unique city values that reside in the state of California and order the results in descending order by city:

- Solution:

```
SELECT DISTINCT city
FROM suppliers
WHERE state = 'California'
ORDER BY city DESC;
```

- These are the results that you should see:

city
Westlake Village
Redwood City
Mountain View

Practice Exercise #1.4: (update)

Q4: Based on the suppliers table populated with the following data, update the city to 'Boise' and the state to "Idaho" for all records whose supplier_name is "Microsoft".

Solution: UPDATE suppliers SET city = 'Boise',

state = 'Idaho' WHERE supplier_name = 'Microsoft';

The supplier table would now look like this

supplier_id	supplier_name	city	state
100	Microsoft	**Boise**	**Idaho**
200	Google	Mountain View	California
300	Oracle	Redwood City	California
400	Kimberly-Clark	Irving	Texas
500	Tyson Foods	Springdale	Arkansas
600	SC Johnson	Racine	Wisconsin
700	Dole Food Company	Westlake Village	California
800	Flowers Foods	Thomasville	Georgia
900	Electronic Arts	Redwood City	California

Practice Exercise #1.5: (delete)

Q5: Based on the suppliers table, delete the supplier record whose state is 'California' and supplier_name is not Google:

Solution: DELETE FROM suppliers WHERE state = 'California'
AND supplier_name <> 'Google';

There would be 3 records deleted and the suppliers table would now look like this:

supplier_id	supplier_name	city	state
100	Microsoft	Redmond	Washington
200	Google	Mountain View	California
400	Kimberly-Clark	Irving	Texas
500	Tyson Foods	Springdale	Arkansas
600	SC Johnson	Racine	Wisconsin
800	Flowers Foods	Thomasville	Georgia

Practice Exercise #2

Sample *employees* table for the queries:

```
CREATE TABLE employees
( employee_number int NOT NULL,
  last_name char(50) NOT NULL,
  first_name char(50) NOT NULL,
  salary int,
  dept_id int, emailid varchar(100),
  CONSTRAINT employees_pk PRIMARY KEY (employee_number)
);
```

Practice Exercise #2:


```
INSERT INTO employees
(employee_number, last_name, first_name, salary, dept_id, emailid)
VALUES
(1001, 'Smith', 'John', 62000, 500, ‘smith@gmail.com’),
(1002, 'Anderson', 'Jane', 57500, 500, ‘anderson@gmail.com’),
(1003, 'Everest', 'Brad', 71000, 501, ‘everest@ny.com’),
(1004, 'Horvath', 'Jack', 42000, 501, ‘horvath@yahoo.com’);
(1005, ‘Mohan', ‘Radha’, 55000, 502, ‘mohan@yahoo.com’);
```

Practice Exercise #2:

1. Select all fields from the *employees* table whose salary is less than or equal to $52,500 (no sorting is required)
2. Select the details of the employee whose first name starts with B.
3. Print name and Id of employee whose email-id is in Gmail.
4. Select the details of the employee who work either for department 501 or 502.
5. Select all the employees working in department 501 in descending order of their salary.

Solution for Practice Exercise #2

- Solution 1:

  SELECT * FROM employees WHERE salary <= 52500;

- Output:

employee_number	last_name	first_name	Salary	dept_id	**Emailid**
1004	Horvath	Jack	42000	501	horvath@yahoo.com

- Solution 2:

  SELECT * FROM employees WHERE first_name like ‘b%’

- Output:-

employee_number	last_name	first_name	salary	dept_id	**emailid**
1003	Everest	Brad	71000	501	Everest@ny.com

Solution for Practice Exercise #2

- Solution 3:

  SELECT employee_number as E_no, last_name, first_name FROM employees where emailid like '%@gmail.com';

- Output

E_no	last_name	first_name
1001	smith	John
1002	Anderson	Jano

- Solution 4:

SELECT * FROM employees WHERE dept_id=501 or dept_id=502; or

SELECT * FROM employees WHERE dept_id in (501, 502)

Output:-

employee_number	last_name	first_name	salary	dept_id	**emailid**
1003	Everest	Brad	71000	501	Everest@ny.com
1004	Horvath	Jack	42000	501	Horvath@yahoo.com
1005	Mohan	Radha	55000	502	mohan@yahoo.com

Solution for Practice Exercise #2

- Solution 5:
  SELECT * FROM employees WHERE dept=501 ORDER BY salary DESC;

- Output

employee_number	last_name	first_name	salary	dept_id	**emailid**
1003	Everest	Brad	71000	501	Everest@ny.com
1004	Horvath	Jack	42000	501	Horvath@yahoo.com

Practice Exercise #3

Sample *customers* table:

Cust_id	First_name	Last_name	Gender	Phone_number
100	Steven	Austin	M	515.123.4567
101	Neena	Singh	F	515.124.4568
102	Bruce	King	M	516.124.4569
103	David	Russell	M	590.123.4560
104	Den	Lee	M	590.124.4561
105	John	Gates	M	590.423.4565
106	Amit	Banda	M	515.124.4550
107	Sundar	Bhatt	M	590.124.4566
108	Clara	Doran	F	515.123.4562

Practice Exercise #3:

1. Write a query to update the portion of the phone_number in the customers table, within the phone number the substring '124' will be replaced by '999’.
2. Write a query to get the details of the customers where the length of the first name greater than or equal to 6.
3. Write a query to extract the last 4 character of phone numbers.
4. Write a query that displays the first name and the length of the first name for all customers whose name starts with the letters ‘D', 'J' or ‘N'. Give each column an appropriate label. Sort the results by the employees' first names.
5. Write a query to display the length of first name for employees where last name contain character ‘a' after 2nd position.

Solution Practice Exercise #3:

- Solution Q1:

```
UPDATE customers SET phone_number = REPLACE(phone_number, '124',
'999')
WHERE phone_number LIKE '%124%';
```

- Solution Q2:

```
SELECT * FROM customers WHERE LENGTH(first_name) >= 6;
```

- Solution Q3:

```
SELECT SUBSTRING(phone_number, 9, 4) as 'Ph.no' FROM customers;
```

- Solution Q4:

```
SELECT first_name "Name", LENGTH(first_name) "Length" FROM customers
WHERE first_name LIKE 'D%' OR first_name LIKE 'J%' OR first_name LIKE
'N%' ORDER BY first_name ;
```

- Solution Q5:

```
SELECT first_name, last_name FROM customers WHERE
INSTR(last_name,'A') > 2;
```