**EXERCISE 1**

Part 1: Creating Natural Joins.

1. Display all of the information about sales representatives and their addresses using a natural join.

SELECT *

FROM sales_representatives NATURAL JOIN sales_rep_addresses;

2. Adapt the query from the previous question to only show the id, first name, last name, address line 1, address line 2, city, email and phone_number for the sales representatives.

SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number

FROM sales_representatives NATURAL JOIN sales_rep_addresses;

Part 2: Creating Joins with the USING Clause

1. Adapt the previous query answer to use the USING clause instead of a natural join.

SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number

FROM sales_representatives JOIN sales_rep_addresses

USING (id);

2. Display all of the information about items and their price history by joining the items and price_history tables.

SELECT *

FROM items JOIN price_history

USING (itm_number);

Part 3: Creating Joins with the ON Clause

Use an ON clause to join the customer and sales representative table so that you display the customer number, customer fist name, customer last name, customer phone number, customer email, sales representative id, sales representative first name, sales representative last name and sales representative email. You will need to use a table alias in your answer as both tables have columns with the same name.

SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email,s.id,s.first_name, s.last_name,s.email

FROM customers c JOIN teams t

ON (t.id = c.team.id);

Part 4- Creating Three-Way Joins with the ON Clause

Using the answer to Task 3 add a join that will allow the team name that the customer represents to be included in the results.

SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email,s.id,s.first_name, s.last_name,s.email, t.name “Team Name”

FROM customers c JOIN sales_representative s

ON (s.id = c.sre_id)

JOIN teams t

ON (t.id = c.tem_id);

Part 5: Applying Additional Conditions to a Join

Using the answer to Task 4 add an additional condition to only show the results for the customer that has the number - c00001.

SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email,s.id,s.first_name, s.last_name,s.email, t.name “Team Name”

FROM customers c JOIN sales_representative s

ON (s.id = c.sre_id)

JOIN teams t

ON (t.id = c.tem_id);

WHERE c.ctr_number =’c00001’;

Part 6: Retrieving Records with Nonequijoins

Write a query that will display name and cost of the item with the number im01101045 on the 12th of December 2016. The output of the query should look like this: The cost of the under shirt on this day was 14.99

SELECT ‘The cost of the’ ||i.name||’on this day was’||p.price AS “Item Details”

FROM items I JOIN price_history p

ON (’12-Dec-2016’ BETWEEN p.start_date AND p.end_date) AND (i.itm_number = ‘im01101045’);

FROM customers CROSS JOIN sales_representatives;

**EXERCISE 2**

Part 1 : Use a Self-Join to Join a Table to Itself (S6L9 Objective 2)

1. Write a query that will display who the supervisor is for each of the sales representatives. The information should be displayed in two columns, the first column will be the first name and last name of the sales representative and the second will be the first name and last name of the supervisor. The column aliases should be Rep and Supervisor.

```
SELECT r.first_name||''||r.last_name AS “Rep”, s.first_name||''|| s.last_name AS “Supervisor”

FROM sales_representatives r JOIN sales_representatives s

ON (r.supervisor_id = s.id);
```

Part 2 : Use OUTER joins (S6L9 Objective 3)

1. Write a query that will display all of the team and customer information even if there is no match with the table on the left (team).

```
SELECT *

FROM teams t RIGHT OUTER JOIN customers c

ON (t.id = c.team_id);
```

Part 3 : Generating a Cartesian Product (S6L9 Objective 4)

1. Create a Cartesian product between the customer and sales representative tables

```
SELECT *

FROM customers CROSS JOIN sales representatives;
```