NO DATE

Lab Exercise 4: DML 3 Part I

Part 1: Creating Natural Joins

```
1. SELECT *
   FROM sales_representatives NATURAL JOIN sales_rep_addresses;
2. 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. 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. SELECT *
   FROM items JOIN price_history
   USING (itm_number);
```

Part 3: Creating Joins with the ON clause

```
1. SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, r.id, r.first_name,
          r.last_name, r.email
   FROM customers c JOIN sales_representatives r
   ON (c.sre_id = r.id);
```

Part 4: Creating Three-Way Joins with the ON clause

```
1. SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, r.id,
          r.first_name, r.last_name, r.email
   FROM customers c JOIN sales_representatives r
   ON (c.sre_id = r.id)
   JOIN teams t
   ON (c.tem_id = t.id);
```

Part 5: Applying Additional Conditions to a Join

```
1. SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, r.id,
          r.first_name, r.last_name, r.email
   FROM customers c JOIN sales_representatives r
   ON (c.sre_id = r.id)
   JOIN teams t
   ON (c.tem_id = t.id)
   WHERE c.ctr_number = 'C00001';
```

Part 6: Returning Records with Nonequijoins

```
1. SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price || As 'Information'
   FROM items i JOIN price_history p
   ON (i.itm_number = p.itm_number)
   WHERE p.start_date '12-DEC-2016' AND p.end_date > '12-DEC-2016';
```

NO. DATE

DML 3 Part 2

Part 1 : Use a self-join to join a Table to itself

```
1. SELECT r.first_name || ' ' || r.last_name REP, s.first_name || ' ' || s.last_name
        SUPERVISOR
   FROM sales_representatives r JOIN sales_representatives s
   ON   (r.supervisor_id = s.id);
```

Part 2: Use OUTER Joins

```
1. SELECT *
   FROM teams RIGHT OUTER JOIN customers
   ON   (id = team_id);
```

Part 3: Generating a cartesian product

```
1. SELECT *
   FROM customers, sales_representatives;
```