Date:

DB Lab 4

Name : Muhammad Firdaus bin Mohideen Abdul Cader
Matrix Num : B23CS0056
Section : 02

SQL 第 4 - DML3 PART 1

part 1 : Create 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 : create join with Using clause.

1. SELECT id, first_name, last_name, addres_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 : create join with ON clause.

1. SELECT c.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 sales_representatives s
   ON (c.sre_id = s.id);

part 4 : create three-way join with ON clause.

1. SELECT c.ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id,
   s.first_name, s.last_none, s.email, t.name
   FROM customers c JOIN sales_representatives s
   ON c.sre_id = s.id
   JOIN teams t
   ON c.tem_id = t.id;

Date:

part 5: Apply additional condition to a join.

```
1. SELECT c.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
   From customers c JOIN sales_representatives s
   ON c.sre_id = s.id
   JOIN teams t
   ON c.tem_id = t.id
   AND c.ctr_number = 'c00001';
```

part 6: Retrieving records with nonequijoins

```
1. SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price
   AS item_cost
   FROM items i JOIN price_history p
   ON i.itm_number = p.itm_number
   JOIN inventory_list il
   ON il.id = i.ilt_id
   AND i.itm_number = 'im01101045' AND p.start_date <= '12 Dec 16'
   AND p.end_date >= '12 Dec 16';
```

SQL 4 - DML 3 PART 2

part 1: self-join

```
1. SELECT rep.first_name || ' ' || rep.last_name "Rep", super.first_name || ' '
   || super.last_name "Supervisor"
   FROM sales_representatives rep JOIN sales_representatives super
   ON rep.supervisor_id = super.id;
```

part 2: Outer joins (Right / Left / Full)

```
1. SELECT *
   FROM teams t FULL OUTER JOIN customers c
   ON t.id = c.tem_id;
```

part 3: Generate a cartesian product.

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