LAB 4 UMAYRA BT SYAMSUL AZHAR B23CS0076

Part 1 : Creating natural joins

1) SELECT *
FROM sales_representatives
Natural JOIN address;

2) SELECT sale-rep-rep.id ~~sale-rep-id~~, sale-rep-firstname, sale-rep-lastname, a-add-line1, a-addline2, cit
sale-rep-email, sale.rep-phoneNo.
FROM sales-representatives sale-rep
Natural JOIN address a;

Part 2 : Creating joins with the USING Clause

1) SELECT rep-id, firstname, lastname, add-line1, add-line2, city, email, phoneNo
FROM sales-representatives
JOIN address USING (rep-id);

2) SELECT i.items-id, i.items-name, i-category, i.description, ph.price, ph start-date,
ph.end-date
FROM items i
JOIN price.history ph ON i.item-id ~~i.items~~ = ph.item-id;

Part 3: creating joins with the ON Clause

1) SELECT c.customer_no,
c.firstName AS customer_firstname,
c.lastName AS customer_lastname,
c.phoneNO AS customer-phoneNo,
c.email AS customer-email,
sr.firstName AS sale_rep_id,

```
sr.firstName  AS  sale-rep-firstname,
sr.lastName  AS  sale-rep-lastname,
sr.email  AS  Sale_email,
FROM  customer c
JOIN  sales-representatives  sr  ON  c.sales_rep.id = sr.rep.id,
```

Part 4. Creating three-way joins with the ON clause

```
1) SELECT  c.customer-number,
           c.firstName  AS  customer_firstname,
           c.lastName  AS  customer_lastname,
           c.phoneNo  AS  Customer-phoneNo,
           c.email  AS  Customer_email,
           sr.rep-id  AS  sale-rep-id,
             sr.firstName  AS  Sale-rep-firstname,
             sr.lastName  AS  Sale-rep-lastname,
             sr.email  AS  Sale-rep-email,
             t.teams_name
   FROM  customer c
   JOIN  sales-representatives  Sr  ON  c.salesrep-id = sr.rep_id,
         teams t  ON  c.teams-id = t.team-id;
```

Part 5' Applying additional conditions to join

```
1) SELECT   c.customer_number,
            c.firstName   AS   customer_firstname,
            c.lastName    AS   customer.lastName,
            c.phoneNO     AS   customer.phoneNo,
            c.email       AS   customer.email,
            sr.rep-id     AS   sale-rep-id,
            sr.firstName  AS   sale-rep-firstname,
            sr.lastName   AS   sale-rep-lastname,
            sr.email      AS   sale-rep-email,
            t.teams_name
FROM   customer c
JOIN   sales.representatives  sr   ON  c.sales_rep-id  = sr.rep-id,
       teams  t  ON  c.teams_id = t.team_id
WHERE
       c.customer_number  =  'c0001';
```

Part 6: retrieving records with Nonequijoins

```
SELECT  i.item_name   AS  item_name,
        ph.price      AS  cost
FROM    items i
JOIN    price_history ph  ON  i.item_id = ph.item.id
WHERE   i.item_number  =  'im01101045'
        AND  ph.start_date  <=  '12-12-2016'
        AND  (ph.end-date IS NULL OR  ph.end_date >= '12-12-2016');
```

LAB 4 PART 2

PART 1 . use a self join to join a table to itself.

```
SELECT Rep.firstname    AS  Rep_firstName,
       Rep.lastname     AS  Rep_last Name,
       Supervisor.firstname  AS  Supervisor_firstName,
       Supervisor.lastname   AS  Supervisor_lastName
FROM   sales.representatives Rep
JOIN   sales_representatives Supervisor ON Rep.superisor_id = Supervisor.rep_id;
```

Part 2 : use OUTER joins

```
SELECT  t.team_name,
        t.team_id,
        c.customer.number,
        c.firstName AS customer_first.name,
        c.lastName  AS customer_last.name,
        c.phoneNo   AS customer_phone.number,
        c.email     AS customer_email
FROM    teams t.
LEFT JOIN customer c ON t.team_id = c.team_id;
```

Part 3

```
SELECT c.customer_number,
       c.firstname  AS customer_firstName,
       c.lastname   AS customer_lastName,
       c.phoneNO    AS customer_phone_number,
       c.email      AS customer_email,
       sr.rep_id    AS sales_rep_id,
       sr.firstname AS sales_rep_firstName,
       sr.lastname  AS sales_rep_lastName
```

```
sr email AS sales.rep_email
FROM customer c
CROSS JOIN sales_representatives sr,
```