```
SELECT * FROM customers;
SELECT * FROM stores;
SELECT * FROM products;
SELECT * FROM orders;
SELECT * FROM shipments;
SELECT * FROM inventory;
SELECT * FROM order_items;

truncate table order_items;
/************************** 2. Split full name into
first name and last name **************************/
--ex: FIRST_NAME: Tammy , LAST_NAME: Bryant

SELECT *
FROM customers
WHERE FIRST_NAME IS NOT NULL
OR LAST_NAME     IS NOT NULL;


SELECT FULL_NAME,
  SUBSTR(FULL_NAME, 1, INSTR(FULL_NAME, ' ')-1) AS
FIRST_NAME,
  SUBSTR(FULL_NAME, INSTR(FULL_NAME, ' ')   +1) AS
LAST_NAME
FROM CUSTOMERS;

UPDATE CUSTOMERS
SET FIRST_NAME = SUBSTR(FULL_NAME, 1, INSTR(FULL_NAME, '
')-1) ,
  LAST_NAME    = SUBSTR(FULL_NAME, INSTR(FULL_NAME, ' ')
  +1);

COMMIT;
```

```
/************************** 3. Correct phone numbers and
email which are not in proper format
**************************/

--Append .com in email id.

SELECT *
FROM CUSTOMERS
WHERE upper(EMAIL_ADDRESS) LIKE upper('%.com%');

UPDATE CUSTOMERS
SET EMAIL_ADDRESS = EMAIL_ADDRESS || '.com'
WHERE upper(EMAIL_ADDRESS) NOT LIKE upper('%.com%');

COMMIT;

--How many records have . in contact number?
--Solution
--remove .* from contact number

SELECT COUNT(*)
FROM CUSTOMERS
WHERE CONTACT_NUMBER LIKE '%.%';

SELECT CONTACT_NUMBER, SUBSTR(CONTACT_NUMBER, 1,
INSTR(CONTACT_NUMBER, '.')-1)
FROM CUSTOMERS
WHERE CONTACT_NUMBER LIKE '%.%';

UPDATE CUSTOMERS
SET CONTACT_NUMBER = SUBSTR(CONTACT_NUMBER, 1,
INSTR(CONTACT_NUMBER, '.')-1)
WHERE CONTACT_NUMBER LIKE '%.%';
```

```
COMMIT;


/************************** 4. Correct contact number
and remove full name **************************/

--How many contact number are less than 10 digit?
--Solution
--Make contact number as 9999999999 if the length is
less than 10.

SELECT CONTACT_NUMBER, LENGTH(CONTACT_NUMBER)
FROM customers
WHERE LENGTH(CONTACT_NUMBER) < 10;

UPDATE CUSTOMERS
SET CONTACT_NUMBER = 9999999999
WHERE LENGTH(CONTACT_NUMBER) < 10;

COMMIT;

--Check for contact number where length is more than 10.

SELECT CONTACT_NUMBER, LENGTH(CONTACT_NUMBER)
FROM customers
WHERE LENGTH(CONTACT_NUMBER) > 10;

SELECT distinct LENGTH(CONTACT_NUMBER)
FROM customers;

--Remove Full Name column from customers table

ALTER TABLE customers DROP COLUMN FULL_NAME;
```

```
/************************** 5. Read BLOB column and fetch attribute details from regular tag
************************/

/*
"colour":"green",
"gender":"Women's",
"brand":"FLEETMIX",
"description":"Excepteur anim adipisicing aliqua ad. Ex aliquip ad tempor cupidatat dolore ipsum ex anim Lorem aute amet.",
"sizes":[0,2,4,6,8,10,12,14,16,18,20],
"reviews":[{"rating":8,"review":"Laborum ipsum adipisicing magna nulla tempor incididunt."},{"rating":10,"review":"Cupidatat dolore nulla pariatur quis quis."},{"rating":9,"review":"Pariatur mollit dolor in deserunt cillum consectetur."},{"rating":3,"review":"Dolore occaecat mollit id ad aliqua irure reprehenderit amet eiusmod pariatur."},{"rating":10,"review":"Est pariatur et qui minim velit non consectetur sint fugiat ad."},{"rating":6,"review":"Et pariatur ipsum eu qui."},{"rating":6,"review":"Voluptate labore irure cupidatat mollit irure quis fugiat enim laborum consectetur officia sunt."},{"rating":8,"review":"Irure elit do et elit aute veniam proident sunt."},{"rating":8,"review":"Aute mollit proident id veniam occaecat dolore mollit dolore nostrud."}]}
*/

/*
```

```
TABLE_NAME, json_table (
    TABLE_NAME.BLOB_COLUMN_NAME
    columns (
      DYNAMIC_COLUMN_NAME_1 DATA_TYPE path
'$.ATTRIBUTE_NAME1',
      DYNAMIC_COLUMN_NAME_2 DATA_TYPE path
'$.ATTRIBUTE_NAME2'
    )
    )
*/

SELECT PRODUCT_ID,
  PRODUCT_NAME,
  UNIT_PRICE,
  PRODUCT_DETAILS,
  COLOUR_NAME,
  GENDER_TYPE
FROM PRODUCTS,
     JSON_TABLE
      (
        PRODUCTS.PRODUCT_DETAILS
        COLUMNS
            ( COLOUR_NAME VARCHAR2(50) PATH '$.colour',
              GENDER_TYPE VARCHAR2(20) PATH '$.gender' )

        );


/************************** 6. Read BLOB column and
fetch attribute details from nested columns
**************************/


/*
```

```
TABLE_NAME, json_table (
    TABLE_NAME.BLOB_COLUMN_NAME, '$.TAG_NAME[*]'
    columns (
      DYNAMIC_COLUMN_NAME_1 DATA_TYPE path
'$.ATTRIBUTE_NAME1',
      DYNAMIC_COLUMN_NAME_2 DATA_TYPE path
'$.ATTRIBUTE_NAME2'
    )
    )
*/

SELECT PRODUCT_ID, PRODUCT_NAME, UNIT_PRICE,
PRODUCT_DETAILS, RATING, REVIEWS
FROM PRODUCTS,
  json_table (
    PRODUCTS.PRODUCT_DETAILS, '$.reviews[*]'
    COLUMNS (
        RATING NUMBER PATH '$.rating',
        REVIEWS VARCHAR2(200) PATH '$.review'
      )
);




/****************** 7. Create separate tables for with
blob attributes ************************/

/*
"colour":"green",
"gender":"Women's",
"brand":"FLEETMIX",
"description":"Excepteur anim adipisicing aliqua ad. Ex
aliquip ad tempor cupidatat dolore ipsum ex anim Lorem
```

```
aute amet.",
"sizes":[0,2,4,6,8,10,12,14,16,18,20],
"reviews":[{"rating":8,"review":"Laborum ipsum
adipisicing magna nulla tempor
incididunt."},{"rating":10,"review":"Cupidatat dolore
nulla pariatur quis
quis."},{"rating":9,"review":"Pariatur mollit dolor in
deserunt cillum
consectetur."},{"rating":3,"review":"Dolore occaecat
mollit id ad aliqua irure reprehenderit amet eiusmod
pariatur."},{"rating":10,"review":"Est pariatur et qui
minim velit non consectetur sint fugiat
ad."},{"rating":6,"review":"Et pariatur ipsum eu
qui."},{"rating":6,"review":"Voluptate labore irure
cupidatat mollit irure quis fugiat enim laborum
consectetur officia sunt."},{"rating":8,"review":"Irure
elit do et elit aute veniam proident
sunt."},{"rating":8,"review":"Aute mollit proident id
veniam occaecat dolore mollit dolore nostrud."}]}
*/

--Create table for regular columns
CREATE TABLE PRODUCT_DETAILS AS
SELECT PRODUCT_ID,
  PRODUCT_NAME,
  UNIT_PRICE,
  COLOUR_NAME,
  GENDER_TYPE,
  BRAND,
  DESCRIPTION,
  SIZES
FROM PRODUCTS,
     JSON_TABLE
      (
```

```
        PRODUCTS.PRODUCT_DETAILS
        COLUMNS
            ( COLOUR_NAME VARCHAR2(50) PATH '$.colour',
              GENDER_TYPE VARCHAR2(20) PATH '$.gender',
              brand,
              description,
              sizes FORMAT JSON
              )
        );

SELECT * FROM PRODUCT_DETAILS;



--Create table for nested columns
CREATE TABLE PRODUCT_RATING AS
SELECT PRODUCT_ID, RATING, REVIEWS
FROM PRODUCTS,
  json_table (
    PRODUCTS.PRODUCT_DETAILS, '$.reviews[*]'
    COLUMNS (
        RATING NUMBER PATH '$.rating',
        REVIEWS VARCHAR2(200) PATH '$.review'
      )
);

SELECT * FROM PRODUCT_RATING;

--Join both the tables
SELECT * FROM
PRODUCT_DETAILS LEFT JOIN PRODUCT_RATING
ON PRODUCT_DETAILS.PRODUCT_ID =
PRODUCT_RATING.PRODUCT_ID
```

```
/****************** 8. Remove invalid records from
order_items where shipment_id is not mapped
************************/


SELECT COUNT(*)
FROM order_items
WHERE SHIPMENT_ID IS NULL;

SELECT COUNT(*) FROM order_items;

SELECT * FROM order_items
WHERE SHIPMENT_ID IS NULL;

DELETE FROM ORDER_ITEMS
WHERE SHIPMENT_ID IS NULL;

COMMIT;


/****************** 9. Map missing first name and last
name with email id credentials
************************/

SELECT *
FROM customers
WHERE FIRST_NAME IS NULL
OR LAST_NAME IS NULL;

SELECT *
FROM CUSTOMERS;

--andrea.james@internalmail.com;
```

```
--james@internalmail.com;

SELECT EMAIL_ADDRESS,
  SUBSTR(EMAIL_ADDRESS, 1, INSTR(EMAIL_ADDRESS, '.')-1)
FIRST_NAME,
  SUBSTR(EMAIL_ADDRESS, INSTR(EMAIL_ADDRESS, '.')+1,
                        INSTR(SUBSTR(EMAIL_ADDRESS,
INSTR(EMAIL_ADDRESS, '.')+1), '@')-1) LAST_NAME,
  INSTR(EMAIL_ADDRESS, '.')+1,
--andrea.james@internalmail.com;
  INSTR(EMAIL_ADDRESS, '@'),
  INSTR(SUBSTR(EMAIL_ADDRESS, INSTR(EMAIL_ADDRESS,
'.')+1), '@')-1 -- STRING: james@internalmail.com;
FROM customers
WHERE FIRST_NAME IS NULL
OR LAST_NAME IS NULL;


--Update data
UPDATE CUSTOMERS
SET FIRST_NAME = SUBSTR(EMAIL_ADDRESS, 1,
INSTR(EMAIL_ADDRESS, '.')-1),
    LAST_NAME = SUBSTR(EMAIL_ADDRESS,
INSTR(EMAIL_ADDRESS, '.')+1,
                       INSTR(SUBSTR(EMAIL_ADDRESS,
INSTR(EMAIL_ADDRESS, '.')+1), '@')-1)
WHERE FIRST_NAME IS NULL
OR LAST_NAME IS NULL;

COMMIT;
```