1. Create a list of products and their total sales. Sort by sales. Which product has the highest total unit sales for each year?

1.a Create a list of products and their total sales. Sort by sales.

```
SELECT PRODUCT.DESCRIPTION, SUM(SALES_FACT.DOLLAR_SALES) AS TOTAL_SALE

FROM STORE_DW.PRODUCT JOIN STORE_DW.SALES_FACT

USING (PRODUCT_KEY)

GROUP BY (PRODUCT.DESCRIPTION)

ORDER BY TOTAL_SALE DESC;
```

Rows returned: 20

1.b Which product has the highest total unit sales for each year?

```
SELECT PRODUCT.DESCRIPTION, SUM(SALES_FACT.DOLLAR_SALES)  AS TOTAL_SALE,
SUM(SALES_FACT.UNIT_SALES) AS TOTAL_UNIT_SALE, TIME.YEAR

FROM STORE_DW.PRODUCT JOIN STORE_DW.SALES_FACT

USING (PRODUCT_KEY)

JOIN STORE_DW.TIME

USING(TIME_KEY)

GROUP BY PRODUCT.DESCRIPTION, TIME.YEAR

ORDER BY TOTAL_UNIT_SALE DESC;
```

Rows returned: 40

Answer: For 1995 Lots of Nuts had the highest total unit sales (14688), and for 1994 Onion Slices had the highest unit sales (14670).

2. Using the answer from the query above, do these products sell better during a particular quarter of each year and is it the same quarter for each year?

```
SELECT PRODUCT.DESCRIPTION, SUM(SALES_FACT.DOLLAR_SALES)  AS TOTAL_SALE,
SUM(SALES_FACT.UNIT_SALES) AS TOTAL_UNIT_SALE,     TO_CHAR(TIME.SQL_DATE,        'YY---
MON')

FROM STORE_DW.PRODUCT JOIN STORE_DW.SALES_FACT

USING (PRODUCT_KEY)

JOIN STORE_DW.TIME

USING(TIME_KEY)

GROUP BY PRODUCT.DESCRIPTION, TO_CHAR(TIME.SQL_DATE,    'YY---MON')

ORDER BY TOTAL_UNIT_SALE DESC;


Rows returned: 120
```

Answer: Lots of Nuts sell better during the fourth quarter of each year. It is the same quarter (i.e, quarter 4 or Oct, Nov, Dec).

3. Create a list of product categories and their total sales. Sort by sales. Does the product category for the highest total unit sales correspond to the answer in 1 above?

Anwer:

```
SELECT PRODUCT.CATEGORY, SUM(SALES_FACT.DOLLAR_SALES) AS TOTAL_SALE

FROM STORE_DW.PRODUCT JOIN STORE_DW.SALES_FACT

USING (PRODUCT_KEY)

GROUP BY (PRODUCT.CATEGORY)

ORDER BY TOTAL_SALE DESC;


CATEGORY       TOTAL_SALE

--------------- ----------

Food           528760.95

Supplies        129494.04

Drinks          123148.6
```

To find whether Lots of Nuts falls under Food we execute following query.

```
SELECT PRODUCT.CATEGORY, PRODUCT.DESCRIPTION, SUM(SALES_FACT.DOLLAR_SALES) AS
TOTAL_SALE
```

```
FROM STORE_DW.PRODUCT JOIN STORE_DW.SALES_FACT
USING (PRODUCT_KEY)
GROUP BY (PRODUCT.CATEGORY, PRODUCT.DESCRIPTION)
ORDER BY TOTAL_SALE DESC;
```

Rows returned: 20

Answer: Lots of Nuts is a type of food. Therefore yes the product category for the highest total unit sales correspond to the answer in 1 above.

4. Which promotion has brought in the most unit sales?

Ans:

```
SELECT PROMOTION.PROMOTION_NAME, SUM(SALES_FACT.DOLLAR_SALES) AS TOTAL_SALE
FROM STORE_DW.PROMOTION JOIN STORE_DW.SALES_FACT
USING (PROMOTION_KEY)
JOIN STORE_DW.PRODUCT
USING (PRODUCT_KEY)
GROUP BY (PROMOTION.PROMOTION_NAME)
ORDER BY TOTAL_SALE DESC;
```

Rows returned: 11

```
PROMOTION_NAME         TOTAL_SALE
------------------------ ----------
No Promotion           579288.14
POS Grabbers            65020.68
```

5. Using the highest and lowest unit sales promotion, is there a pattern in the products between the promotions?

Ans:

```
SELECT PROMOTION.PROMOTION_NAME, PRODUCT.DESCRIPTION, SUM(SALES_FACT.UNIT_SALES) AS
TOTAL_UNIT_SALE

FROM STORE_DW.PROMOTION JOIN STORE_DW.SALES_FACT

USING (PROMOTION_KEY)

JOIN STORE_DW.PRODUCT

USING (PRODUCT_KEY)

GROUP BY (PROMOTION.PROMOTION_NAME, PRODUCT.DESCRIPTION)

ORDER BY TOTAL_UNIT_SALE DESC;
```

Rows returned: 55

Answer: We did not find any particular pattern.

6. Create an interesting query by joining tables together, aggregating data in the fact table, and using the dimensions for drill downs and roll ups. You must use GROUP BY. This query must use 3 or more dimensions.

```
/* Find the store and its sale that sold 'Lots of Nuts' using 'Ads and Racks' as promotion

during October and display the result in decreasing order of Sales */


SELECT  s.name AS "Store Name", SUM(f.unit_sales) AS Sales

FROM store_dw.product p

JOIN store_dw.sales_fact f USING (product_key)

JOIN store_dw.promotion pm USING (promotion_key)

JOIN store_dw.store s USING (store_key)

JOIN store_dw.time t USING (time_key)

WHERE p.description = 'Lots of Nuts'

AND to_char(t.sql_date,'MON') = 'OCT'

AND pm.promotion_name = 'Ads and Racks'

GROUP BY s.name, to_char(t.sql_date,    'MON')

ORDER BY SUM(f.unit_sales) DESC;
```

Rows returned: 14