Scalable Data Engineering, Exercise 3

Henry Haustein

Task 1

(a) True

(b) False, maintenance is different

(c) False, it is an NP-hard problem

(d) False, it depends on the use case

(e) True

Task 2

(a) SQL:

```
1 EXPLAIN(SELECT * FROM ORDERS)
```

(b) SQL:

```
1 SELECT * FROM ORDERS WHERE O_CLERK = "Clerk#000000322"
```

took 600ms.

(c) SQL:

```
1 CREATE INDEX ocleak_idx ON ODERS(O_CLERK)
```

(d) Query from (b) then takes around 300ms

Task 3

(a) SQL:

```
1 CREATE MATERIALIZED VIEW custperreg AS
2   SELECT REGION.R_NAME, COUNT(*)
3   FROM CUSTOMER, NATION, REGION
4   WHERE
5     CUSTOMER.C_NATIONKEY = NATION.N_NATIONKEY AND
6     NATION.N_REGIONKEY = REGION.R_REGIONKEY
7   GROUP BY REGION.R_NAME
8 WITH DATA
```

(b) SQL:

```
INSERT INTO REGION(R_REGIONKEY, R_NAME, R_COMMENT) VALUES (5,
    "AUSTRALIA", "down under")

REFRESH MATERIALIZED VIEW custperreg
```

(c) In the materialized view Australia is missing. That's why there are no customers for this region and since we are doing inner joins for the materialized view there are no joining partners for Australia. To fix this:

```
DROP MATERIALIZED VIEW custperreg

CREATE MATERIALIZED VIEW custperreg AS
  SELECT REGION.R_NAME, COUNT(CUSTOMER.C_CUSTKEY)
  FROM
    CUSTOMER RIGHT OUTER JOIN (
      NATION RIGHT OUTER JOIN REGION
      ON(NATION.N_REGIONKEY = REGION.R_REGIONKEY)
    )
    ON(CUSTOMER.C_NATIONKEY = NATION.N_NATIONKEY)
  GROUP BY REGION.R_NAME
WITH DATA
```

Task 4

(a) We create a materialized view from the subquery:

```
CREATE MATERIALIZED VIEW linepart AS
  SELECT
    PART.P_NAME,
    SUM(LINEITEM.L_QUANTITY) AS qty
  FROM PART, LINEITEM
  WHERE PART.P_NAME = LINEITEM.L_PARTKEY
  GROUP BY PART.P_NAME
WITH DATA

SELECT PART.P_NAME,  qty,  RANK() OVER(ORDER BY qty DESC)
FROM linepart
```

Creating the materialized view takes some time but then the actual query is done in about a second (from 10 seconds before).

(b) SQL:

```
CREATE MATERIALIZED VIEW linepart2 AS
  SELECT
    PART.P_NAME,
    SUM(LINEITEM.L_QUANTITY) AS qty,
    LINEITEM.L_SUPPKEY
  FROM PART, LINEITEM
  WHERE PART.P_NAME = LINEITEM.L_PARTKEY
  GROUP BY PART.P_NAME, LINEITEM.L_SUPPKEY
```

```
9  WITH DATA
```

The query from (a) becomes then

```
1  SELECT PART.P_NAME,  SUM(qty),  RANK() OVER(ORDER BY SUM(qty)
       DESC)
2  FROM linepart2
```

with roughly 4 seconds run time. The other query becomes

```
1  SELECT
2    PART.P_NAME,
3    NATION.N_NAME,
4    SUM(qty),
5    RANK() OVER(
6      PARTITION BY PART.P_NAME
7      ORDER BY SUM(qty) DESC
8    )
9  FROM linepart2, NATION, SUPPLIER
10 WHERE
11   LINEITEM.L_SUPPKEY = SUPPLIER.S_SUPPKEY AND
12   SUPPLIER.S_NATIONKEY = NATION.N_NATIONKEY
13 GROUP BY PART.P_NAME, NATION.N_NAME
```

with now 21 seconds runtime (was 60 seconds before).