

! record solution5.rpt

-- For the order clerks (O_CLERK) Clerk#000000522, Clerk#000000154, find the
-- total number of ordered parts per customer (O_CUSTKEY), per supplier
-- (L_SUPPKEY), per customer and supplier (O_CUSTKEY, L_SUPPKEY), and the total
-- number of ordered parts.

```
SELECT O_CUSTKEY, L_SUPPKEY, COUNT(*)
FROM LINEITEM JOIN ORDERS
      ON LINEITEM.L_ORDERKEY = ORDERS.O_ORDERKEY
WHERE O_CLERK IN ('Clerk#000000522', 'Clerk#000000154')
GROUP BY O_CUSTKEY, L_SUPPKEY WITH CUBE;
```

-- For the parts with the keys (L_PARTKEY) 7, 8, 9 find the largest discount
applied
-- (L_DISCOUNT) per part key (L_PARTKEY) and per part key and supplier key
-- (L_PARTKEY, L_SUPPKEY) and the largest discount applied at all.

```
SELECT L_PARTKEY, L_SUPPKEY, MAX(L_DISCOUNT)
FROM LINEITEM
WHERE L_PARTKEY IN (7, 8, 9)
GROUP BY L_PARTKEY, L_SUPPKEY WITH ROLLUP;
```

-- Find the smallest price (L_EXTENDEDPRICE) per order year (O_ORDERDATE), and
-- order clerk (O_CLERK).

```
SELECT O_CLERK, substr(O_ORDERDATE,1,4), MIN(L_EXTENDEDPRICE)
FROM ORDERS JOIN LINEITEM
      ON LINEITEM.L_ORDERKEY = ORDERS.O_ORDERKEY
GROUP BY O_CLERK, substr(O_ORDERDATE,1,4)
GROUPING SETS ( (O_CLERK), (substr(O_ORDERDATE,1,4)) );
```

-- For each part list its key (PS_PARTKEY), all its available quantities
(PS_AVAILQTY),
-- the smallest available quantity, and the average available quantity. Consider
only the
-- parts with the keys 5 and 15.

```
SELECT PS_PARTKEY, PS_AVAILQTY, MIN(PS_AVAILQTY) OVER (PARTITION BY PS_PARTKEY),
      AVG(PS_AVAILQTY) OVER (PARTITION BY PS_PARTKEY)
FROM PARTSUPP
WHERE PS_PARTKEY IN (5,15);
```

-- For each part list its key (PS_PARTKEY) and all its available quantities
-- (PS_AVAILQTY) sorted in descending order and a rank (position number in an
-- ascending order) of each quantity. Consider only the parts with the keys 10 and
20. Use
-- an analytic function ROW_NUMBER().

```
SELECT PS_PARTKEY, PS_AVAILQTY, ROW_NUMBER() OVER (PARTITION BY PS_PARTKEY
                                                    ORDER BY PS_AVAILQTY DESC)
FROM PARTSUPP
WHERE PS_PARTKEY IN (10,20);
```

```
-- For each part list its key (PS_PARTKEY), its available quantity, and an average
available
-- quantity (PS_AVAILQTY) of the current quantity and all previous quantities in
the
-- ascending order of available quantities. Consider only the parts with the keys
15 and
-- 25. Use ROWS UNBOUNDED PRECEDING sub-clause within PARTITION BY
-- clause.
```

```
SELECT PS_PARTKEY, PS_AVAILQTY, AVG(PS_AVAILQTY) OVER (PARTITION BY PS_PARTKEY
                                                         ORDER BY PS_AVAILQTY
                                                         ROWS UNBOUNDED PRECEDING)
```

```
FROM PARTSUPP
WHERE PS_PARTKEY IN (15,25);
! record
```