ADVANCE QUERY

1. Menampilkan Nama Member, Nama supplier, nama barang, tanggal pemesanan yang pernah memesan

QUERY :

SELECT

m.nama AS NAMAMEMBER,

s.nama AS NAMASUPLIER,

b.nama\_barang AS NAMABARANG,

p.tanggal\_pemesanan AS TANGGALPEMESANAN

FROM pemesanan p

JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan

JOIN member m ON p.memberid = m.memberid

JOIN barang b ON pd.kode\_barang = b.kode\_barang

JOIN suplier s ON s.suplierid = b.suplierid

ORDER BY m.memberid



2. Menampilkan data supplier yang belum pernah melakukan suplai barang

QUERY:

SELECT

s.suplierid,

s.nama,

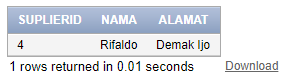
s.alamat

FROM barang b

RIGHT JOIN suplier s ON s.suplierid = b.suplierid

WHERE b.suplierid IS NULL

ORDER BY s.suplierid



3. Menampilkan Nama Member yang memesan lebih dari 1 barang

QUERY:

SELECT \*

FROM (SELECT

m.nama AS NAMA,

COUNT(pd.kode\_barang) AS JUMLAHBARANG

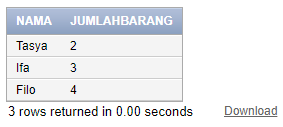
FROM pemesanan p

JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan

JOIN member m ON p.memberid = m.memberid

GROUP BY m.nama)

WHERE jumlahbarang > 1



4. Menampilkan Nama Member, jumlah barang yang dibeli dan total harga dari barang yang dipesan

QUERY:

SELECT

m.nama,

COUNT(pd.kode\_pemesanan) AS JUMLAHBARANG,

(COUNT(pd.kode\_pemesanan) \* SUM(b.harga)) AS TOTALHARGA

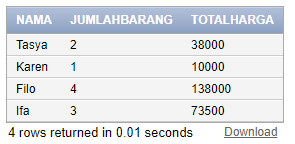
FROM pemesanan p

JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan

JOIN member m ON p.memberid = m.memberid

JOIN barang b ON b.kode\_barang = pd.kode\_barang

GROUP BY m.nama



VIEW

1. VDATAPEMESANAN

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW  "VDATAPEMESANAN" ("NAMAMEMBER", "NAMASUPLIER", "NAMABARANG", "TANGGALPEMESANAN") AS    SELECT     m.nama AS NAMAMEMBER,    s.nama AS NAMASUPLIER,    b.nama\_barang AS NAMABARANG,    p.tanggal\_pemesanan AS TANGGALPEMESANAN FROM pemesanan p JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan JOIN member m ON p.memberid = m.memberid JOIN barang b ON pd.kode\_barang = b.kode\_barang JOIN suplier s ON s.suplierid = b.suplierid ORDER BY m.memberid;​ |

2. VJUMLAHPEMBELIAN

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW  "VJUMLAHPEMBELIAN" ("NAMA", "JUMLAHBARANG") AS    SELECT    m.nama AS NAMA,   COUNT(pd.kode\_barang) AS JUMLAHBARANG FROM pemesanan p JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan JOIN member m ON p.memberid = m.memberid GROUP BY m.nama ORDER BY JUMLAHBARANG;​ |

3. VTOTALPEMESANAN

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW  "VTOTALPEMESANAN" ("NAMA", "JUMLAHBARANG", "TOTALHARGA") AS    SELECT     m.nama,    COUNT(pd.kode\_pemesanan) AS JUMLAHBARANG,    (COUNT(pd.kode\_pemesanan) \* SUM(b.harga)) AS TOTALHARGA FROM pemesanan p JOIN pemesanandetail pd ON p.kode\_pemesanan = pd.kode\_pemesanan JOIN member m ON p.memberid = m.memberid JOIN barang b ON b.kode\_barang = pd.kode\_barang GROUP BY m.nama;​ |

4. VNULLSUPLIERBARANG

|  |
| --- |
| CREATE OR REPLACE FORCE VIEW  "VNULLSUPLIERBARANG" ("ID", "NAMA SUPLIER", "ALAMAT") AS    SELECT    s.suplierid,   s.nama,   s.alamat FROM barang b RIGHT JOIN suplier s ON s.suplierid = b.suplierid WHERE b.suplierid IS NULL ORDER BY s.suplierid;​ |

TRIGGER

1. TGDELETE\_HISTORY

|  |
| --- |
| create or replace trigger "TGDELETE\_HISTORY" BEFORE delete on "BARANG" for each row begin INSERT INTO HISTORY\_BARANG VALUES (    :old.kode\_barang,    :old.nama\_barang,    :old.harga,    :old.stock,    :old.suplierid,    CURRENT\_TIMESTAMP,    'DELETED'); end;​ |

2. TGINSERT\_HISTORY

|  |
| --- |
| create or replace trigger "TGINSERT\_HISTORY" AFTER insert on "BARANG" for each row begin INSERT INTO HISTORY\_BARANG VALUES (    :new.kode\_barang,    :new.nama\_barang,    :new.harga,    :new.stock,    :new.suplierid,    CURRENT\_TIMESTAMP,    'INSERTED'); end;​ |

3. TGUPDATE\_HISTORY

|  |
| --- |
| create or replace trigger "TGUPDATE\_HISTORY" BEFORE update on "BARANG" for each row begin INSERT INTO HISTORY\_BARANG VALUES (    :old.kode\_barang,    :old.nama\_barang,    :old.harga,    :old.stock,    :old.suplierid,    CURRENT\_TIMESTAMP,    'UPDATED'); end;​ |

4. TGUPDATE\_STOCK

|  |
| --- |
| create or replace trigger "TGUPDATE\_STOCK" AFTER insert on "PEMESANANDETAIL" for each row DECLARE stok number; begin select stock into stok from barang where kode\_barang = :new.kode\_barang; stok := stok - :new.jumlah; UPDATE barang set stock=stok where kode\_barang = :new.kode\_barang; end;​ |