

1)

a)

```
select land, count(*) as Anzahl_Kunden  
from kunden  
Group by land;
```

b)

```
select land, count(*) as Anzahl_Kunden  
from kunden  
group by land  
having count(*) >= 5  
order by Anzahl_Kunden desc;
```

c)

```
select land, count(distinct ort) as Anzahl_Orte  
from kunden  
group by land;
```

d)

```
select coalesce(artikel.ArtikelNr, 0) as ArtikelNr,  
coalesce(artikel.ArtikelName, 0) as ArtikelName,  
coalesce(sum(ap.Bestellmenge), 0) as Menge,  
coalesce(sum(ap.Verkaufspreis * ap.Bestellmenge), 0) as Umsatz  
from auftragspositionen as ap  
right join artikel on ap.ArtikelNr = artikel.ArtikelNr  
where artikel.Massstab = "1:18"  
group by artikel.ArtikelNr  
order by Menge asc;
```

e)

```
select kunden.land as land, count(distinct auftragspositionen.ArtikelNr) as AnzahlArtikel  
from kunden  
join auftraege on kunden.KundenNr = auftraege.KundenNr  
join auftragspositionen on auftraege.AuftragsNr = auftragspositionen.AuftragsNr  
where kunden.land in ("Deutschland", "Frankreich", "England")  
group by kunden.land;
```

f)

```
select kunden.KundenNr, kunden.Firma,  
sum(auftragspositionen.Verkaufspreis*auftragspositionen.Bestellmenge) as Umsatz,  
(sum(auftragspositionen.Verkaufspreis*auftragspositionen.Bestellmenge) - sum(artikel.Einkaufspreis  
* auftragspositionen.Bestellmenge)) as Gewinn, count(auftragspositionen.AuftragsNr) as  
AnzahlAufträge  
  
from kunden join auftraege on kunden.KundenNr = auftraege.KundenNr  
  
join auftragspositionen on auftraege.AuftragsNr = auftragspositionen.AuftragsNr  
  
join artikel on auftragspositionen.ArtikelNr = artikel.ArtikelNr  
  
group by kunden.KundenNr  
  
having Umsatz > 100000  
  
order by Umsatz asc;
```

2)

a)

Drop view ArtikelUmsatz;

Create View ArtikelUmsatz as

```
Select ArtikelNr, sum(Bestellmenge) as Bestellmenge, sum(Verkaufspreis*Bestellmenge) as Umsatz  
from auftragspositionen  
  
group by ArtikelNr;
```

b)

Eine erneute Aggregation auf der View ist nicht erforderlich, da die Aggregationen schon in der View gespeichert sind.

3)

a)

Man müsste für jeden erzeugten Auftrag prüfen ob es eine passende Auftragsposition gibt. Dies kann man nicht durch eine referenzielle Integritätsbedingung sicherstellen, da diese nur Referenzen auf eine Tabelle angibt, Man müsste einen Trigger schreiben, der aktiviert wird wenn die Auftrags Tabelle geupdated wird und somit der Trigger die Auftragspositions Tabelle auch updated.

b)

Select *

from auftraege, auftragspositionen

where auftraege.AuftragsNr Not In (Select auftragsnr from auftragspositionen);

c)

Select *

```
from auftraege join auftragspositionen on not(auftraege.AuftragsNr = auftragspositionen.AuftragsNr)
where auftraege.AuftragsNr Not in (select AuftragsNr from auftragspositionen);
```