Bewertung vom Produkt:

SELECT count(Bewertung) as 'Anzahl Bewertungen', round(AVG(Bewertung),2) as 'Bewertung (1-10)' FROM Bestellung where ArtikelID=4

select(SELECT Verkaufspreis FROM Artikel WHERE ArtikelID=1) \* (SELECT count(ArtikelID) FROM Gekauft WHERE ArtikelID=1) from dual;

**Datensätze die älter als 30 Tage sind mit dem Vermerk: in Bearbeitung**

delete from Bestellung where Bestelldatum < NOW() - INTERVAL 30 DAY and BestellstatusID = 1

**Wenn man auf Button BESTELLEN drückt, ändert sich der Bestellstatus auf BESTELLT**

UPDATE Bestellung set BestellstatusID=2 where AuftragsNummer=<AUFTRAGSNUMMER DER BESTELLUNG>

**Artikel Bewerten die man gekauft hat**

UPDATE Bestellung SET Bewertung=7 where

ArtikelID=4 and

KundenID=3 and

AuftragsNummer = 1002 and

BestellstatusID != 1

**Für Garantiefälle, so sieht man alle Bestellte Artikel von einem Kunden**

SELECT a.ArtikelBeschreibung, b.BestellstatusID, k.Vorname, k.Nachname from

Bestellungen b join Artikel a on b.ArtikelID=a.ArtikelID join Kunde k on b.KundenID=k.KundenID where b.KundenID=1

**Artikel kaufen**

INSERT INTO `Bestellung` (`AuftragsNummer`, `ArtikelID`, `KundenID`) VALUES

(1003,7,4),

(1003,9,4);

**Alle Artikel der Oberkategorie ‘KLEIDER’ anzeigen:**

SELECT ArtikelBeschreibung, Verkaufspreis, UnterkategorieID FROM Artikel WHERE UnterkategorieID=3 or UnterkategorieID = 4

**Alle Artikel der Oberkategorie ‘ELEKTRONISCHE GERÄTE’ anzeigen:**

SELECT ArtikelBeschreibung, Verkaufspreis, UnterkategorieID FROM Artikel WHERE UnterkategorieID=1 or UnterkategorieID = 2

**Alle Artikel der UNTERKATEGORIE 1-4 anzeigen**

SELECT ArtikelBeschreibung, Verkaufspreis FROM Artikel WHERE UnterkategorieID=1,2,3,4

**Verkaufspreis von einem Artikel um 20% senken:**

select round((Verkaufspreis\*0.8),2) as 'Artikel 1 um 20% reduzieren' from Artikel where ArtikelID = 1

**Verkaufspreis vom Artikel multipliziert mit der Anzahl verkauften Artikeln**

select count(b.ArtikelID), (count(b.ArtikelID)\*a.Verkaufspreis) from Bestellungen b right JOIN Artikel a on b.ArtikelID=a.ArtikelID where a.ArtikelID=7 and b.BestellstatusID != 1

**Verkaufszahlen aller Artikeln addiert von der Oberkategorie ‘ELEKTRONISCHE GERÄTE’ inkl. Schüler/Händler Rabatt)**

select round(sum(a.Verkaufspreis\*p.PreisReduktion),2) as 'Verkauf der Kategorie ELEKTRONISCHE GERÄTE in CHF' from Artikel a join Bestellungen b on b.ArtikelID=a.ArtikelID join Kunde k on k.KundenID=b.KundenID join Preisgruppen p on p.PreisgruppenID=k.PreisgruppenID where a.UnterkategorieID=1 or a.UnterkategorieID=2 and b.BestellstatusID!=1

**Alle Artikel mit Einkaufspreis, Verkaufspreis (Endkunde,Schüler,Händler):**

select a.ArtikelID, a.Einkaufspreis, a.Verkaufspreis as 'Endkundenpreis', round((a.Verkaufspreis\*0.8),2) as 'Schülerpreis', round((a.Verkaufspreis\*0.85),2) as 'Händlerpreis' from Artikel a

**Für einzelne Preisgruppen Preise aller Artikel abfragen**

select a.ArtikelID, a.Einkaufspreis, a.Verkaufspreis, round((a.Verkaufspreis\*p.PreisReduktion),2) from Artikel a, Preisgruppen p where p.PreisgruppenID=1

**Umsätze aller Artikel**

select ArtikelBeschreibung, round((Verkaufspreis-Einkaufspreis),2) as 'Umsatz pro Artikel' from Artikel

**Grösster Umsatz von allen Artikeln**

select ArtikelBeschreibung, max(round((Verkaufspreis-Einkaufspreis),2)) as 'Umsatz pro Artikel' from Artikel

**Kleinster Umsatz von allen Artikeln**

select ArtikelBeschreibung, min(round((Verkaufspreis-Einkaufspreis),2)) as 'Umsatz pro Artikel' from Artikel

**Durchschnittlicher Umsatz aller Artikeln**

select COUNT(ArtikelBeschreibung), avg(round((Verkaufspreis-Einkaufspreis),2)) as 'Durchschnittsumsatz' from Artikel

**Umsatz Kategorie Apple**

select sum(round((Verkaufspreis-Einkaufspreis),2)) as 'Durchschnittsumsatz Kategorie Apple' from Artikel where UnterkategorieID=1

**Umsatz Preisgruppe Schüler**

select round(sum((a.Verkaufspreis\*p.PreisReduktion)-a.Einkaufspreis),2) as 'Umsatz der Preisgruppe Schüler' from Artikel a join Bestellungen b on a.ArtikelID=b.ArtikelID join Kunde k on b.KundenID=k.KundenID join Preisgruppen p on p.PreisgruppenID=k.PreisgruppenID where p.PreisgruppenID=1

**Umsatz Preisgruppe Endkunde**

select round(sum((a.Verkaufspreis\*p.PreisReduktion)-a.Einkaufspreis),2) as 'Umsatz der Preisgruppe Endkunde' from Artikel a join Bestellungen b on a.ArtikelID=b.ArtikelID join Kunde k on b.KundenID=k.KundenID join Preisgruppen p on p.PreisgruppenID=k.PreisgruppenID where p.PreisgruppenID=2

**Umsatz Preisgruppe Händler**

select round(sum((a.Verkaufspreis\*p.PreisReduktion)-a.Einkaufspreis),2) as 'Umsatz der Preisgruppe Endkunde' from Artikel a join Bestellungen b on a.ArtikelID=b.ArtikelID join Kunde k on b.KundenID=k.KundenID join Preisgruppen p on p.PreisgruppenID=k.PreisgruppenID where p.PreisgruppenID=3